

AD-A061 380

STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB
BCA AND HRA: TWO PROGRAMS FOR COMPUTING ECONOMIC EQUILIBRIA. (U)
AUG 78 T ELKEN

F/G 5/3

UNCLASSIFIED

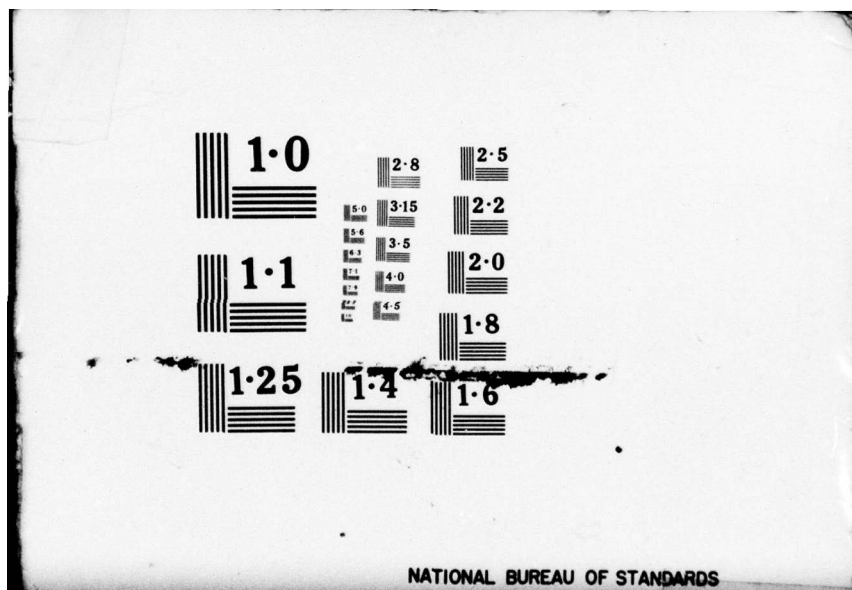
SOL-78-17

N00014-75-C-0267

NL

1 OF 2
ADA
061380

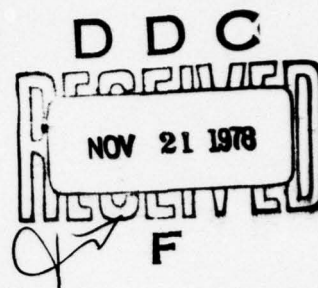






Systems
Optimization
Laboratory

(12)
B.S.



A061380

LEVEL



DDC FILE COPY

THIS DOCUMENT IS BEST QUALITY REPRODUCTION.
THE COPY FURNISHED TO DDC CONTAINS A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

This document has been approved
for public release and sale; its
distribution is unlimited.

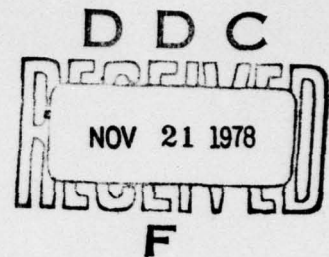
Department of Operations Research
Stanford University
Stanford, CA 94305

78 11 16 002

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DDC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
Stanford University
Stanford, California
94305



BCA AND HRA: TWO PROGRAMS FOR
COMPUTING ECONOMIC EQUILIBRIA

by

Thomas Elken

TECHNICAL REPORT SOL 78-17
August 1978

Research and reproduction of this report were partially supported by the U.S. Department of Energy Contract EY-76-S-03-0326 PA #18; The Office of Naval Research Contract N00014-75-C-0267; and the National Science Foundation Grants MCS76-20019 A01 and MCS76-81259 A01.

Reproduction in whole or in part is permitted for any purposes of the United States Government. This document has been approved for public release and sale; its distribution is unlimited.

78 11 16 002

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
I	Introduction	1
II	The Bilinear Complementarity Algorithm (BCA)	7
	II.1. Input Requirements	7
	II.2. Main Program	15
	II.3. Subroutines of BCA	21
	II.4. Sample Problems	29
	II.5. Source Listing for BCA	44
III	The HRA (Homotopy Retraction Algorithm) Code for Solving Equilibrium Problems	79
	III.1. Revisions to the Original Code	79
	III.2. Input Requirements	82
	III.3. Main Program	83
	III.4. Subroutines of HRA	85
	III.5. Sample Problems	89
	III.6. HRA Source Listing	96
	References	128

ACCESSION for

Wife Section ☒

B H Section ☐

NTIS ☐

DDC ☐

DISPATCHED TO: ACTIVITY CODES

SPECIAL

A 23
E.H.

BCA and HRA: Two Programs for
Computing Economic Equilibria

by
Thomas Elken

I. Introduction

BCA and HRA are computer programs designed to solve a version of the economic equilibrium problem. For an introduction to this type of problem and models for dealing with it, the reader is directed to Elken [2]. Very briefly, we will describe mathematically the problem which these codes solve.

Find x , t , λ , and ζ such that

$$Dx + t = b$$

$$\lambda D - \zeta = e_{\text{NCOL}}$$

$$\langle C_i, \lambda \rangle - \lambda_i t_i = 0, \quad i = 1, \dots, \text{IH} \quad (1)$$

$$\lambda_i t_i = 0, \quad i = \text{IH} + 1, \dots, \text{NROW}$$

$$\langle x, \zeta \rangle = 0$$

$$x, t, \lambda, \zeta \geq 0,$$

where D is a matrix with $NROW$ rows and $NCOL$ columns, IH is the number of consumers or households, e_{NCOL} is the $NCOL$ th unit vector, and $\langle \cdot, \cdot \rangle$ is the usual inner product.

The BCA is an implementation of the bilinear complementarity algorithm presented in Wilson [7] and in Elken [2]. The HRA code implements the homotopy retraction algorithm as described in [2]. The reader must consult the latter work to be able to formulate a problem correctly and to be cognizant of the conditions under which these algorithms will solve his/her problem.

The algorithms have so many features in common that many of the subroutines in the two programs are identical. In particular, both algorithms begin by solving the linear program

$$\begin{aligned} &\text{maximize } x_{NROW} \\ &\text{subject to } [I|D] \begin{bmatrix} t \\ x \end{bmatrix} = b, \\ &t, x \geq 0. \end{aligned} \tag{2}$$

To solve this linear program, the code LPML written by John Tomlin is used. Actually, the subroutines comprising LPML are part of both codes, BCA and HRA. LPML has been documented in J. Tomlin [4]. LPML has been converted into a linear complementarity code, LCPL, which has been documented in J. Tomlin [5] and [6]. Since

LCPL uses many of the subroutines of LPM1 without change, the reader is referred to the works cited for a description of how the matrix $A = [I|D]$ is stored and how transformations are processed.

Next we discuss some important details of the implementation of BCA and HRA.

Source Language

The programs are written entirely in FORTRAN IV for the IBM series 360 and 370 computers. They are WATFIV compatible.

Specification

The main program is executed as a job step. The program input is described in Section II.1. All input occurs in subroutines MAIN, INPUT, and BASINP.

Error Indicators

Error indicators and other diagnostics and messages will be indicated in the documentation for each subroutine. The error indicators for the LPM1 portion of the code are described in [4].

Subroutines

The routines making up BCA and HRA are as follows:

BCA	HRA
MAIN	MAIN
BLCONS	BLCONS
FINDP	FINDP
PIVOT	PIVOT
UPAKC	UPAKC
BSCNG	BSCNG
SUPERB	SUPERB
RECALC	RECALC
ENDPNT	ENDPNT
QUADS	QUADS
DSENT	DSENT
	GTN
	DERIVG
CONCHK	CONCHK
FTN	FTN
DERIV	DERIV
NORM	NORM
DECOMP	DECOMP
SOLVE	SOLVE
SING	SING
DEBUG	DEBUG

The dotted lines indicate the subroutines which are essentially identical.

The remainder of the subroutines are from LPML for both programs.

BLOCK DATA

INPUT

FTRAN

BTRAN

FORMC

TRUEDJ

PRICE

CHUZR

WRETA

SHIFTR

INVERT

UNPACK

SHFTE

CHSOL

UPBETA

NORMAL

ITEROP

UNRAVL

CRASH

BASINP

CLEAR

These subroutines are described in greater detail in the remaining sections of this report.

Program Size

The total length of BCA is 3831 source statements (including comments and common blocks, etc.); HRA is 3772 statements long. LPM1 comprises 1819 statements of each of these.

Array Size

The standard version of the codes requires a blank common array of 133,256 bytes of core. This is dependent on the setting of maximum problem dimensions.

There are eleven labelled common blocks which require 97,448 bytes of core. Only the size of common blocks LNCONS and INDXZ are dependent upon problem dimensions.

Accuracy and Convergence

The linear constraints (1) and (2) are satisfied with a tolerance of ZTOLZE with a default of $1.E - 4$. The bilinear constraints (3) are satisfied to a tolerance of TOLFZ; $1.E - 10$ is the default.

Convergence is guaranteed in the BCA and HRA when the first IH values of b are chosen appropriately. See Elken [2] Chapters V-VI for a discussion of the convergence properties of these algorithms and suggestions for improvements.

Timing

Using the FORTRAN H compiler with $OPT = 2$ the codes have solved a problem with 6 consumers and 4 goods with $D \in R^{23 \times 20}$ in .45 and .38 seconds for the BCA and HRA, respectively. A problem with 3 consumers and 56 goods and $D \in R^{97 \times 107}$ required 4.54 and 6.19 seconds for the BCA and HRA to solve. A description of numerical results for these and several more test problems is in Chapter V of [2].

II. The Bilinear Complementarity Algorithm (BCA)

II.1. Input Requirements

The program input consists of 4 (or 5) segments.

- A) Parameter specification relevant to equilibrium problem
- B) Parameter specification relevant to the auxiliary linear program
- C) Data for the LP in MPS standard form
- D) Basis input (optional, see [5])
- E) Specification of the C matrix (3). (Initial endowments for the individual consumers.)

We will now deal with these in succession.

A. Program Parameter Input

Parameters are changed from their default value for each problem by means of a FORTRAN "NAMELIST" input from the card reader. The first card must start with &PARM1 in column 2 or later; each card must begin with a blank, and the list must be terminated by &END.

Example

columns

1 2 3

&PARM1 IH = 3, L = 2, ICECHO = 1, &END

The parameters which can be defined by the &PARM1 statement are listed below along with their default values.

REAL *4 Tolerances

TOLFZ (DEFAULT = 1.E - 10)

Termination criterion for the solution of the bilinear equations.

TOLBD (DEFAULT = 1.E - 4)

Feasibility tolerance for the linear constraints.

TOLCV (DEFAULT = 1.E - 5)

Termination criterion for the return to the curve.

INTEGER *4 Parameters

IH Number of consumers (traders). This parameter has no default.

It must be specified in the &PARM1 statement.

ICNTRL (DEFAULT = 1)

If ICNTRL.LT.0, then the program behaves just like LPM1 to solve one or more linear programs. This may be useful when one desires to solve some preliminary linear programs to determine initial utility levels. Otherwise, an equilibrium problem will be processed by BCA.

IECHO (DEFAULT = 0)

If IECHO = 1 while ICNTRL.LT.0, the D matrix of (1) will be printed out, ten columns at a time.

ICECHO (DEFAULT = 0)

If .EQ.1, the C-matrix of (3) will be printed.

IPROD (DEFAULT = 0)

If .EQ.1, it is assumed that the consumers own certain shares of the producing firms. Thus, data containing those shares must appear following the LP data in 10F7.4 format.

ITLIME (DEFAULT = 500)

The maximum number of cells which we allow the program to pass through.

KOUTB (DEFAULT = 0)

Unit number of output of the optimal basis for the auxiliary linear program. If KOUTB.EQ.0, no basis will be saved.

REAL *4 Parameters

STPMX (DEFAULT = 100)

The maximum possible step in the direction tangent to the curve.

STPRD (DEFAULT = 0.5)

The factor between 0 and 1 used to reduce the steplength when it exceeds STPMX.

B. Linear Program Parameters

Next, a list containing parameters of importance for the linear programming portion of the code must be included. The beginning of this list is &PARAM. We leave the description of these parameters to [4] with one exception. The parameter IOBJ contains the row number of the objective row in the D matrix. Since this row is always e_{NROW}^T (2) no flexibility is necessary here so we require that IOBJ be the last row of D. Since we did not want to alter the LPML part of the code, the user must know how many rows D has and enter the appropriate value for IOBJ in the parameter list. This rather irritating requirement makes bookkeeping easier in many parts of the code.

Also, the auxiliary linear program (4) is a maximization problem while LPM1 is a minimization code by default. This can be remedied in one of three ways:

- 1) Let the last row of D be $-e_{NROW}^T$,
- 2) Set ZSCALE = -1.0 in the &PARAM list, or
- 3) Set IOBJ = -NROW in the &PARAM list.

Remember to do only one of the alternatives above.

C. Problem Input

The D and b referred to in (1) are read from unit KINP (DEFAULT = 5) in slightly modified "MPS format". The card images required are:

NAME Card

This has "NAME" in columns 1 to 4 and the (up to 8 characters) problem name in columns 15-22. This card is optional but highly desirable.

ROWS Card

Has "ROWS" in columns 1-4.

Row Names

Each row is assigned a (unique) name of up to 8 characters, one per card, in columns 5-12. Embedded blanks are allowed (unlike

MPS). Also row types must be supplied in columns 3 of each card

L: \leq constraint

G: \geq constraint

E: = constraint

N: objective function row .

For the equilibrium problem all constraints will be of type "L" except for the objective function.

COLUMNS Card

Has "COLUMNS" in columns 1-7.

Matrix-Element

The non-zero elements of D are supplied by column. All the elements of a column must be together. Each column is assigned a (unique) name of up to eight characters. The format is:

cols	1-4	5-12	15-22	25-36	40-47	50-61
	blank	column	row	element	second	second
		name	name	value	row	element
					name	value
					(optional)	(optional)

Again embedded blanks are allowed in column names.

RHS Card

Has "RHS" in columns 1-3.

Right Hand Side Elements

The right hand side vector (b) may be given a name, which should be different from any row or column name. The elements are given in the same format as the matrix elements.

ENDATA Card

Has "ENDATA" in columns 1-6. Sample input is shown in Section II.4.

D. LP Basis Input

If KINB is initialized to a nonzero value in &PARAM, a basis is read from unit KINB. If KINB.EQ.KINP, the basis must be entered following the LP problem input, and before the ENDATA card. This basis will be the initial basis for solving the auxiliary LP (2).

NAME Card

Has "NAME" in columns 1-4 and the basis name in columns 15-22. If this name does not agree with the problem name, or the name card is missing, a warning is given.

Basis Cards

For basic columns of D, the column names are given (one per card) in columns 5-12. For basic columns of I (slack variables), the corresponding row name is given in columns 5-12.

ENDATA Card

Has ENDATA in columns 1-6.

E. C Matrix Input

In most problems the data for C will be predominately extracted from the RHS vector b. For this reason we use a rather short, but complicated, form for the input of C.

The data is read from unit KINP directly following the MPS format data. The instructions constructed from the pointers in this data produce a matrix packed into a vector in the same manner that $[I|D]$ is packed (see [6]).

In the current version of the code it is assumed that (if $IPROD.NE.0$) each consumer I is associated with a fraction $SHR(I)$, which represents his share of each firm in the economy, such that $\sum_{I=1}^{IH} SHR(I) = 1.0$. If this assumption is not true, the correct C matrix can still be input; however, it requires more work.

The form of the data for inputting the C matrix is as follows

1. One or more cards containing $(SHR(I), I = 1, IH)$ in 10F7.4 format (only if $IPROD.NE.0$).

2. Several cards describing the column of C associated with consumer 1.
3. A blank card.
4. Information associated with consumer 2, etc.

We now elaborate on the specific requirements of point 2 above. Assume that we are dealing with column I of C.

The first card for column I contains values for the integer variables ISHR, LL, and KK in 3I4 format. ISHR is a control parameter which behaves as follows:

- A) If ISHR.EQ.1, the program reads one or more consecutive elements of the RHS corresponding to constraints on one or more of the firms in the economy. These numbers are multiplied by SHR(I), the Ith consumers share of the firm, and stored in C.
- B) If ISHR.EQ.0, the program reads one or more consecutive elements of the RHS corresponding to the constraints on consumer I alone.
- C) If ISHR.EQ.-1, the program reads one or more elements from the card(s) following in the input stream in 10F7.4 format. These usually correspond to commodities (or firms) which are owned by a number of consumers, but not in any fixed proportion.

LL and KK determine the range of rows of C which will be altered due to this card. If only one row is involved, either KK can be set equal to zero (leave columns 9-12 blank), or set KK equal to LL.

A blank card indicates that the program should move on to the next column of C (the next consumer).

If the problem has been formulated and input correctly, the sum of the columns of C must be equal to the RHS vector (see [2], Section V.3). Examples of input and output for some small problems will be given after the program in Section II.4.

II.2. Main Program

The main program consists of 350 statements which perform a variety of functions: the parameters are initialized and altered by reading the name list PARM1, the subroutines of LPML are called to solve the auxiliary linear program, the data is read so that the C matrix can be constructed, the machinery is set up for the bilinear complementarity algorithm to begin, the main path-following subroutine ENDPNT is called, the updates and decisions made necessary by the output from ENDPNT are performed, and the final solution is printed. If options were added so that a number of equilibrium problems could be solved, this main program would undoubtedly become sort of a master subroutine, as NORMAL is for LPML.

Restrictions relevant to the use of BCA.

1. The number of consumers (IH) must be less than or equal to 10.
2. A must have not more than 350 rows or 400 columns.

Of course these limits can be extended by redimensioning all vectors of size 10, 350 or 400 and allocating more core.

Before describing the subroutines of BCA we will define the variables in the labelled common blocks and some of those in blank common (those of use in the subroutines outside of LPM1).

Variable Glossary

We have specified that almost all real variables are REAL*8 by the statement

```
IMPLICIT REAL*8 (A-H, O-Z) .
```

Notable exceptions are the initial data in A and C which are stored as REAL*4 variables.

COMMON/LP1/:

PI(1302) The complete set of dual variables (prices and relative costs).

XX(1302) The complete set of primal variables, slacks first and then the structural variables in their original order.

COMMON/BLCST/ (Bilinear constraints)

COMMON/BLCST2/

These variables store the current version of the budget surplus function in terms of the superbasic variables (see Elken [2] Section III.3). The functional form is

$$\text{Type 1: } f_{\mu} = BF1 + D1*PI(MU) - \text{DIAG}(PI(MU))*(F1*XX(NU) + E1)$$

(3)

$$\text{Type 2: } f_{\nu} = BF2 + D2**PI(MU) - \text{DIAG}(XX(NU))*(F2*PI(MU) + E2) ,$$

where $PI(MU)$ refers to those components $PI(J)$ of PI such that $MU(I) = J$ for some $I = 1, 2, \dots, KMU$, KMU is the current number of dual superbasic variables, and

$$\text{DIAG}(PI(MU)) = \begin{pmatrix} PI(MU(1)) & 0 & \dots & 0 \\ 0 & PI(MU(2)) & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & PI(MU(KMU)) \end{pmatrix}.$$

$XX(NU)$ and KMU are defined similarly.

Also in `COMMON/BLCST/` are `C(800)`, `IC(800)`, and `LC(20)` which store C in packed form in exactly the same manner as A .

`COMMON/LNCONS/`

These matrices and vectors allow us to express the basic variables in terms of the superbasic variables. If IH and $DIGMA$ are index sets for the primal and dual variables, respectively, then the relationship is

$$XX(JH(I)) = \sum_{J=1}^{KNU} G1(I,J) * XX(NU(J)) + BA(I) \quad i = 1, \dots, M$$

$$PI(DIGMA(I)) = \sum_{J=1}^{KMU} G2(I,J) * PI(MU(J)) + BB(I) \quad I = 1, \dots, N,$$

where the arrays are dimensioned G1(350,10), G2(400,10), BA(350), BB(400).

COMMON/INDX1/NUH(10), MUH(10), NU(10), MU(10)

NUH(I) = $\begin{cases} 0 & \text{if } XX(I) \text{ is not superbasic} \\ J & \text{if } XX(I) \text{ is the } J\text{th} \\ & \text{superbasic variable } I = 1, \dots, IH \end{cases}$

MUH(I) = $\begin{cases} 0 & \text{if } PI(I) \text{ is not superbasic} \\ J & \text{if } PI(I) \text{ is the } J\text{th} \\ & \text{dual superbasic variable, } I = 1, \dots, IH \end{cases}$

NU(I) = J if the Ith superbasic variable is XX(J)

MU(I) = J if the Ith dual superbasic variable is PI(J).

Note that NUH(NU(I)) = I.

COMMON/INDX2/JH,DIGMA,KINBAS,IDBAS

These arrays correspond to σ , $\bar{\sigma}$, γ and $\bar{\gamma}$ in the description of the algorithm in Chapter III of [2].

JH(I) = J if XX(J) is basic and pivots on row I.

DIGMA(I) = J if PI(J) is dual basic and corresponds to the Ith
row of G2.

$$\text{KINBAS(I)} = \begin{cases} 0 & \text{if XX(I) is non-basic} \\ J & \text{if XX(I) is basic and pivots on row J.} \end{cases}$$
$$\text{IDBAS(I)} = \begin{cases} 0 & \text{if PI(I) is not dual basic} \\ j & \text{if PI(I) is dual basic and corresponds to the Jth} \\ & \text{row of G2.} \end{cases}$$

COMMON/SCAL/

BT = A scalar that helps to define the normal hyperplane subproblem
in ENDPNT.

JJ = The index of the constraint in G1 or G2 which defines the
exiting facet for the current cell.

MFLAG = A flag which defines the type of jacobian which must be
calculated.

DD1 = The number of bilinear constraints which are currently at
zero.

P = The index of the pivot column determined by subroutine FINDP.

PD = The flag which determines whether the initial facet corresponds
to a primal (PD = 1), a dual (PD = -1) variable, or a bilinear
constraint (PD = 0) at zero.

MPD = Identical to PD except that it refers to the final facet of
the current cell.

$$\text{INEQ} = \begin{cases} 1, & \text{if the bilinear inequality is the last of the Type 1} \\ & \text{constraints.} \\ 2, & \text{if the bilinear inequality is the last of the Type 2} \\ & \text{constraints see (1).} \end{cases}$$

KFUN = Counts the number of functionals evaluated in (1).

KJAC = The number of partial derivatives of the functions defined
by (1) which are calculated during the course of the algorithm.

COMMON/DIM/ (DIMENSIONS)

IH = The number of households.

M = Number of rows in the problem.

N = Number of structural variables in the problem.

DD = Current number of superbasic variables (= DD1 + 1).

KMU = Current number of primal superbasics.

KNU = Current number of dual superbasics (DD = KMU + KNU)

$$\text{BL1} = \begin{cases} \text{KMU} - 1, & \text{if } \text{INEQ} = 1 \\ \text{KMU}, & \text{otherwise} \end{cases}$$

$$\text{BL2} = \begin{cases} \text{KNU} - 1, & \text{if } \text{INEQ} = 2 \\ \text{KNU}, & \text{otherwise} \end{cases}$$

MP1 = M + 1

NM = N + M

COMMON/INT/

IPS(30) A vector storing the permutations for a LU decomposition
linear equation solver used in conjunction with Newton's
method.

KDET The sign of the determinant of the current jacobian of f.

KOUNT The number of times ENDPNT has been called.

ISING A flag which is 1 when the current jacobian is singular,
0 otherwise.

COMMON/TOLER/TOLFZ, TOLBD, TOLCV, ICNTRL, IECHO

These parameters are defined above.

II.3. Subroutines of BCA

(2) SUBROUTINE BLCONS: Calculates the coefficients of the first DD
bilinear constraints. DD1 of these are constraints which define
the curve. The last is an inequality which is not yet binding,
but which must be the next one to become binding.

The calculations are motivated by the fact that the bilinear
functions

$$\sum_{j=1}^M C(I,J)*PI(J) - PI(I)*XX(I), \quad I = 1, \dots, DD$$

can be reduced to functions of the superbasic variables by using
(6). The position of DD in the index sets MU and NU deter-
mines BL1, BL2, INP, INQ, and INEQ.

- (3) SUBROUTINE FINDP (PD1, IS, P): Finds the element of largest absolute value in the row determined by (PD1, IS) and stores the index of the variable corresponding to that element in P. When we say the row determined by (PD1, IS) we mean G1(IS,.) if PD1 = 1, and G2(IS,.) if PD1 = -1.
- (4) SUBROUTINE PIVOT (SS,RR): If variable XX(SS) is entering the basis and variable XX(RR) is leaving the basis, the columns of G1 and G2 and the columns BA and BB must be updated. This is accomplished by a simple pivot. The details for this pivot are contained in Section III.5 of [2].
- (5) SUBROUTINE UNPACKC(II): A call to this subroutine causes the IIth column of C to be unpacked and stored in the M-vector Y.
- (6) SUBROUTINE BSCNG(S,R): This subroutine updates the index sets JH, KINBAS, DIGMA, and IDBAS when variable S replaces variable R in the primal basis.
- (7) SUBROUTINE SUPERB(KEY, PD1, IS, PD2,JS): Revises the index sets defining the superbasic variables. It also adds or removes columns of G1 and G2 depending on the values of the parameters

$$KEY = \begin{cases} 0, & \text{if columns are both added and removed,} \\ 1, & \text{if columns are only added,} \\ 2, & \text{if columns are only removed.} \end{cases}$$

(PD1,IS) determine the column to be added.

(PD2,JS) determine the column to be removed.

The calculations involved in calculating the column to be added are described in [2] Section III.3.

- (8) SUBROUTINE RECALC: The parameter INVFRQ of LPM1 specifies how frequently the product form basis is to be reinverted. Each time the basis is reinverted while the BCA portion of the code is in effect, G1, G2, and BA, BB are recomputed using the new basis inverse. The same thing could be accomplished by DD calls to SUPERB with KEY = 1, and (PD1, IS) appropriately specified.
- (9) SUBROUTINE ENDPNT (JS, PD1, IS, NET): implements the path-following algorithm described in Section III.4 [2]. The initial point is in a facet of the cell defined by (6) and the bilinear inequality; we call this the initial facet. The algorithm follows the curve defined by the DD1 bilinear constraints and the DD superbasic variables. The objective is to follow the curve until it intersects another facet of the cell and to find the point where that intersection occurs. We call that point the endpoint which is contained in the final facet. The initial facet is determined by the equation described by (PD1, JS). The final facet is determined by (PD1, IS). NET counts the number of Newton iterations which were required in the various subproblems of the algorithms referred to above.

A common block is referred to for the first time in this subroutine: COMMON/NEWT/H(10, 11), X(10) Z(10), ACC(2, 10). The following vectors are used in this routine: U(3), V1(10), (V2(10) F(10), DOT(4), RHS(10), UL(10, 10).

H(10, 11) stores the jacobian for the various Newton subproblems
 X(10) contains the superbasic variables as $X = (PI(MU), XX(NU))$
 Z(10) is the correction to X supplied by the Newton method,
 $Z = H^{-1}F$
 ACC(2, 10) stores the two vectors which determine the current linear
 approximation to the curve. $f^{-1}(0)$
 U(3) stores the coefficients for any quadratics that have to
 be solved
 V1(10) holds the gradient of the functional defining the initial
 facet
 V2(10) contains a vector in the null space of H , a tangent to
 $f^{-1}(0)$
 F(10) holds the values of f and any other functional involved
 in the current subproblem
 DOT(4) a collection of accumulators
 RHS(10) a work space for finding the tangent vector by solving
 a linear system; $RHS = H(\cdot, ICX)$.
 UL(10, 10) stores the LU decomposition of the current jacobian matrix

Subroutines called: QUADS, DSENT, CONCHK, FTN, DERIV NORM,
 DECOMP, SOLVE.

It should be noted that the Newton's method being implemented
 is a modified Newton's method with descent. The iteration is

$$X^{k+1} = X^k - \alpha(f'(X^k)^{-1} f(X^k)) \quad k = 0, 1, \dots$$

where α is the first scalar of $\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots\}$ such that

$$\|f(X^{k+1})\| < \|f(X^k)\|.$$

The basic loop is something like the following

```

DO 100 K = 1, 15
    CALL FTN(F, X)           calculate f(X)
    CALL NORM (F, S1, DD)    set S1: = ||f(X)||
    IF (S1.LT.TOLCV) GO to 800 terminate if S1 < 10-5
    CALL DERIV              calculate H: = f'(X)
    CALL DECOMP(DD, H, UL)   decompose H = UL
    CALL SOLVE(DD, UL, F, Z) solve by back substitution
                             UL*Z = F

    ALPHA = 1.0
    CALL DSENT (ALPHA, S1)   implements descent on the
                             norm of f.

    DO 95 I = 1, DD
        X(I) = X(I) - ALPHA*Z(I)

95  CONTINUE
100 CONTINUE

C  TAKE CORRECTIVE ACTION IF NEWTONS
C  METHOD HAS NOT TERMINATED WITHIN
C  15 ITERATIONS

```

- (10) SUBROUTINE QUADS (U, IMAG, ALPHA, BETA): finds both real roots of the quadratic

$$U(1) \cdot \alpha^2 + U(2) \cdot \alpha + U(3) = 0 \quad .$$

If there are no real roots, the flag IMAG is set equal to 1. If there is a double root or if $U(1) = 0$, the root is stored in both ALPHA and BETA. If there are two real roots ALPHA stores the smaller root and BETA the larger.

- (11) SUBROUTINE CONCHK (GMIN, KGMIN, MP2): checks whether the constraints of (1) are satisfied at the point defined by the current superbasic values. The basic variables are evaluated in terms of the superbasic variables as suggested by (4). GMIN stores the minimum value of these variables and (MP2, KGMIN) indicate which variable attains this minimum. The theory tells us that we only have to check the current bilinear inequality among the bilinear constraints described in (3). If this value is less than GMIN, MP2 is set equal to zero.

- (12) SUBROUTINE FTN(F, X): evaluates the first DD1 bilinear functionals as they were defined in BLCONS (3). The last component F(DD) stores a different function value depending upon which subproblem is being solved in ENDPNT.

If MFLAG = 0, then $F(DD) = 0$.

If MFLAG = 1, then F(DD) contains the value of the functional specified by (MPD, JJ).

If MFLAG = 2, then F(DD) = X(JJ).

If MFLAG = 3, then F(DD) = $\langle \text{ACC}(2, \cdot), X \rangle - \text{BT}$ (the normal hyperplane to $\text{ACC}(2, \cdot)$, if F(DD) = 0).

- (13) SUBROUTINE DERIV: computes the exact jacobian of the function calculated in f. The form of the jacobian is as follows

$$H = \begin{bmatrix} \text{PI}(\text{MU}) & \text{XX}(\text{NU}) \\ \hline \text{D1} & \text{DIAG}(\text{PI}(\text{MU})) * \text{F1} \\ \hline \text{D2} & -\text{DIAG}(\text{F2} * (\text{PI}(\text{MU}) + \text{E2})) \end{bmatrix} - \begin{bmatrix} \text{PI}(\text{MU}) & \text{XX}(\text{MU}) \\ \hline \text{DIAG}(\text{F1} * \text{XX}(\text{NU}) + \text{E1}) & 0 \\ \hline \text{DIAG}(\text{XX}(\text{NU})) * \text{F2} & 0 \end{bmatrix}$$

The last row of H contains the gradient of the last functional of F as described above.

- (14) SUBROUTINE DSENT (ALPHA, PNORM): Implements descent on the norm of f in Newton's method as described above. ALPHA is cut in half until $\|f(x^{k+1})\| < \text{PNORM} = \|f(x^k)\|$ or until $\text{ALPHA} \leq 10^{-5}$.

- (15) SUBROUTINE DECOMP (NN, A, UL): (Borrowed from Forsythe [3].

The reader should look there for a complete description.)

Implements a Gaussian elimination scheme with partial pivoting to produce a LU decomposition of the matrix A. A few statements were added to this subroutine so that KDET would change sign every time a row interchange was performed. KDET will eventually be the sign of det A (see Elken [1977a], sec. III.2).

- (16) SUBROUTINE SOLVE (NN, UL, B, X): is also borrowed from Forsythe [3] with the additions of statements which change the sign of KDET every time $UL(I, I) < 0$. SOLVE uses backsubstitution to solve the system $L*U*X = B$. L and U are a lower and upper triangular matrix which are stored in the square matrix UL.

At the end of SOLVE, KDET is equal to $\text{sgn}(\det A)$.

- (17) SUBROUTINE SING(IWHY): is also borrowed from [3]. It is called from DECOMP and SOLVE to print messages concerning the singularity of A depending on the parameter IWHY.
- (18) SUBROUTINE DEBUG(MODE): prints information which may be useful in debugging the program when changes are made. The information which is printed depends upon the value of MODE.

MODE = 1: Prints the time left in this job step by calling the system subroutine LEFTLA(TIME) where TIME is a REAL*4 variable. When running this program in WATFIV or on a system other than SLAC, this portion of code should be removed.

MODE = 2: The arrays BA, G1, BB, and G2 will be printed by rows.

MODE = 3: The index sets defining the primal basis, JH and KINBAS will be printed.

MODE = 4: Information about the location and value of the first IH primal and dual variables will be given (KNU, KMU, NVH, MUH, (XX(I), PI(I), I = 1, ..., IH)).

MODE = 5: The coefficients of the bilinear constraints will be printed: BF1, BF2, D1, D2, E1, E2, F1, and F2.

MODE = 6: A variety of variables will be printed out which are of use in debugging ENDPNT: PD, KMU, KNU, INEQ, MUH, NUH, JH, DIGMA, PI and XX.

The remaining 1815 lines of BCA contain the subroutines which comprise LPM1. The reader is referred to [4], [5], and [6] for documentation relating to them.

II.4. Sample Problems

The sample input and output are given below for two small problems. The computational results for these problems are given in [2] Chapter V.

For the reader to learn how to formulate an equilibrium problem so that it can be input and solved by BCA, (or HRA) it is necessary to read Sections II.2 and II.3 of [2]. We will discuss the definitions of D , b and C of (1) for a small problem here.

The problem devised by Mas Colell (Wilson [7] is concerned with three consumers and two goods. By using the theory of Section II.2 of Elken [2] we see that solving the equilibrium problem devised by Mas Colell is equivalent to solving (1) with

$$D = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ .5 & 1. & .25 & 1. \\ 1. & .5 & .20 & 1. \end{bmatrix}, \quad b = \begin{bmatrix} -0.9 \\ -0.95 \\ -3.92 \\ 3.0 \\ 3.0 \end{bmatrix},$$

and

$$C = \begin{bmatrix} -0.9 & 0 & 0 \\ 0 & -0.95 & 0 \\ 0 & 0 & -3.92 \\ 1. & 1. & 1. \\ 1. & 1. & 1. \end{bmatrix}.$$

Notice that the sum of the columns of C is b . This will always be the case. We are almost ready to type the data cards for BCA, but we have to add a row to D to represent the objective row and assign names to the rows and columns. Below is our choice of names.

	ACT1	ACT2	ACT3	MEXP
UTIL1	-1			
UTIL2		-1		
UTIL3			-1	
GOOD1	.5	1.	.25	1.
GOOD2	1.	.5	.20	1.
OBJ				1.

$$D = \left[\begin{array}{c} \text{UTIL1} \\ \text{UTIL2} \\ \text{UTIL3} \\ \text{GOOD1} \\ \text{GOOD2} \\ \text{OBJ} \end{array} \begin{bmatrix} -1 & & & & \\ & -1 & & & \\ & & -1 & & \\ .5 & 1. & .25 & 1. \\ 1. & .5 & .20 & 1. \\ & & & 1. \end{bmatrix} \right]$$

We only enter the nonzero values to emphasize the fact that only the nonzero elements of D and b need be entered.

In the sample input below (Figure 1) we specify that we want

three consumers (IH = 3) ,
two goods (L = 2) ,
no production (IPROD = 0) ,

and a

C matrix input check (ICECHO = 1) .

Notice that we specify the I th column of C by reading the I th element from b and placing it in the corresponding position of $C(\cdot, I)$, and we read in the 4th and 5th elements from the card following
'-1 4 5'.

Sample Output

The output for the solution of the Mas-Colell problem by BCA is on the following pages. The current form of the program is rather verbose -- an option should be put in to print only the final solution if that is all that the user wants to see.

The first page of output shows the values of all parameters, gives the problem name and the output concerning the solution of the auxilliary linear program. Messages concerning time can be ignored in this printout because no timing routine was being called.


```

0.5 MRS-COLELL
1.   &PARAM1 IH=3,L=2,IPOD=0,ICECH=1, &END
2.   &PARAM IOBJ=-6, &END
3.   NAME MRS-COL
4.   ROWS
5.     L UTIL1
6.     L UTIL2
7.     L UTIL3
8.     L GOOD1
9.     L GOOD2
10.    N OBJ
11.   COLUMNS
12.     RCT1 UTIL1 -1.0 GOOD1 0.5
13.     RCT1 GOOD2 1.0
14.     RCT2 UTIL2 -1.0 GOOD1 1.0
15.     RCT2 GOOD2 0.5
16.     RCT3 UTIL3 -1.0 GOOD1 0.25
16.5    RCT3 GOOD2 0.20
17.     MEXP GOOD1 1.0 GOOD2 1.0
18.     MEXP OBJ 1.0
19.   RHS
20.     ENDOW UTIL1 -0.9 UTIL2 -0.95
21.     ENDOW UTIL3 -3.92
22.     ENDOW GOOD1 3.0 GOOD2 3.0
23.   ENDDATA
24.     0 1 0
25.     -1 4 5
26.     1. 1.
27.
28.     0 2 0
29.     -1 4 5
30.     1. 1.
31.
32.     0 3 0
-3.     -1 4 5
4.     1. 1.
=

```

Figure 1

The following pages give information concerning the behavior of the bilinear complementarity algorithm, and finally, the solution. As a measure of the work done by the algorithm, the number of calls to subroutine ENDPNT, the number of scalar function evaluations, and the number of $IX \times IH$ jacobian evaluations are printed. The solution is printed in a common format for linear programs:

JH(I)	The names of the primal variables which may be non-zero in the equilibrium solution
VALUE	gives the value of these variables. The value of MEXP should always be near zero in an equilibrium.
ROW NAMES	is self-explanatory
PI(I)	gives the value of the dual multiplier associated with that row. Those PI's associated with commodity balance rows are the usual "equilibrium prices."
RHS	gives the value of the original RHS vector (b).

56.
57.
58.
59.
60.
61.
62.
63.
64.
65.
66.
67.
68.
69.
70.
71.
72.
73.
74.
75.
76.
77.
78.
79.
80.
81.
82.
83.
84.
85.
86.
87.
88.
89.
90.
91.
92.
93.
94.
95.
96.
97.
98.
99.
100.
101.
102.
103.
104.
105.
106.
107.
108.
109.
110.
111.
112.
113.
114.
115.

```

COLUMNS 1 THROUGH 3
-0.9000 0.0 0.0
0.0 -0.9500 0.0
0.0 0.0 -3.9200
1.0000 1.0000 1.0000
1.0000 1.0000 1.0000
0.0 0.0 0.0

-----
1TH ENDPOINT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
A2(*U)+ A3
1.0000000 0.0 0.442000 CONSTRAINT TYPE:( 1, 4)
STEPLENGTH= 0.442000 F= 0.0
ITERATION 0 X= 0.442000 F= 0.0
NORM(F(X))= 0.0
AFTER 0 NEWTON ITERATIONS.
THE 5TH PRIMAL VARIABLE WENT TO ZERO.
VECIN: UTIL1
VECOU: GCCD2

-----
2TH ENDPOINT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
A2(*U)+ A3
1.0000000 0.245156 0.500000
STEPLENGTH= 0.245156 CONSTRAINT TYPE:( 0, 7)
ITERATION 0 X= 0.74516 F= 0.0
NORM(F(X))= 0.0
AFTER 0 NEWTON ITERATIONS.

THE BILINEAR CONSTRAINT HIT ZERO.
THERE ARE NOW 1 BUDGET SURPLUSES AT ZERO.

-----
3TH ENDPOINT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
A2(*U)+ A3
-0.435442 0.874267 0.7451564
STEPLENGTH= 0.304255 0.0
ITERATION 0 X= 0.59746 0.26600 CONSTRAINT TYPE:( 1, 5)
NORM(F(X))= 0.39248 F= 0.039288 -0.000000
ITERATION 1 X= 0.62185 0.26600 F= 0.0
NORM(F(X))= 0.0

THE QUADRATIC PICKED THE WRONG CONSTRAINT.
WITH A VALUE OF -0.067781
ITERATION 0 X= 0.80332 -0.12552
NORM(F(X))= 0.42621 F= 0.022773 0.425603
ITERATION 1 X= 0.58738 0.22983
NORM(F(X))= 0.108519 F= 0.076734 -0.076734
ITERATION 2 X= 0.64148 0.21569
NORM(F(X))= 0.001082 F= 0.000765 -0.000765
ITERATION 3 X= 0.64203 0.21555
NORM(F(X))= 0.000000 F= 0.000000 -0.000000

```

```

116.. SUPERBASIC INFO: KNU= 1 KNU= 1
117.. NUH(1)= 0 1 0
118.. NUH(1)= 1 0 0
119.. Y = 0.657548 0.215548 0.0
120.. LAMDA= 0.642035 0.857965 0.235797
121.. THE TIME LEFT IS NOW 1.000000 SEC.
122.. AFTER 4 NEWTON ITERATIONS.
123..
124.. THE BILINEAR CONSTRAINT HIT ZERO.
125.. THERE ARE NOW 2 BUDGET SURPLUSES AT ZERC.
126..
127.. -----
128.. 4TH ENDPT CALL:
129.. THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
130.. A2(*U)+ A3
131.. -0.0263976 0.6420346
132.. -0.0358613 0.2155483
133.. 0.5990081 0.0
134.. STEPLENGTH= 0.079954 CONSTRAINT TYPE:( 0, 7)
135.. ITERATION 0 X= 0.63992 0.21268 0.07988
136.. NORM(F(X))= 0.055999 F= 0.00011 0.00006 0.055999
137.. ITERATION 1 X= 0.63394 0.20460 0.20701
138.. NORM(F(X))= 0.000169 F= 0.000088 0.000048 -0.000136
139.. ITERATION 2 X= 0.63397 0.20470 0.20650
140.. NORM(F(X))= 0.00000 F= -0.000000 0.000000 -0.000000
141.. AFTER 2 NEWTON ITERATIONS.
142..
143.. THE BILINEAR CONSTRAINT HIT ZERO.
144.. THERE ARE NOW 3 BUDGET SURPLUSES AT ZERC.
145..
146..
147.. AFTER 4 ENDPT CALLS, WE HAVE EQUILIBRIUM.
148..
149.. TOTALS: SCALAR FUNCTION CALLS= 37
150.. JACOBIAN EVALUATIONS= 4
151..
152.. THE TIME LEFT IS NOW 1.000000 SEC.
153.. UTILITY LEVELS: 1.5773504 1.1547005 4.2264982
154.. PRICES: 0.633975 0.866025 0.236603
155.. MAS-COL
156..
157.. JH(1) VALUE ROW NAMES PI(1) RHS
158.. ACT1 1.57735041 UTIL1 0.63397455 -0.900000000
159.. ACT2 1.15470049 UTIL2 0.86602545 -0.900000000
160.. ACT3 4.22649817 UTIL3 0.23660254 -3.920000000
161.. UTIL1 0.67735041 GOOD1 0.73205090 3.000000000
162.. MEXP -0.00000023 GOOD2 0.26794910 3.000000000
163.. UHJ 0.00000023 OBJ -1.000000000 0.0
164..
165..
166..
167..
168..
169..
170..

```

Sample Problem: Whisman

Next we give a sample input and output for a problem formulated by Al Whisman (Wilson [1976]). The problem involves four traders and three goods. Each trader is concerned with two to four activities. To describe the problem we give only the D, b, and C matrices of problem (1):

$$D = \begin{bmatrix} -1 & -1 & & & & & & & & \\ & & -1 & -1 & & & & & & \\ & & & & -1 & -1 & -1 & -1 & & \\ & & & & & & & & -1 & -1 \\ 1 & 2 & 4 & 3 & & 1 & & 3 & 1 & 2 & 1 \\ 2 & 2 & 3 & 2 & 1 & 2 & 3 & & 1 & & 1 \\ 3 & 2 & 1 & 2 & & & 4 & 3 & & & 1 \\ & & & & & & & & & & 1 \end{bmatrix},$$

$$b^T = [-.48 \quad -.32 \quad -5.1 \quad -.95 \quad 5 \quad 11 \quad 9],$$

$$C = \begin{bmatrix} -.48 & & & \\ & -.32 & & \\ & & -.51 & \\ & & & -.95 \\ 2 & 1 & 1 & 1 \\ 2 & 3 & 5 & 1 \\ 1 & 3 & 4 & 1 \end{bmatrix}.$$

Again, only the non-zero elements of the matrices are given.

The input follows. The first card, again, is just a header;
it is not to be included in the input stream.

The sample output for the Whisman problem also follows, page
40.

WHISMAN

\$PARAM IH=4,L=3,IPROD=0,ICECHO=1,\$END

\$PARAM ICBJ=-8,\$END

NAME WHISMAN

ROWS

104. L UTIL1
105. L UTIL2
106. L UTIL3
107. L UTIL4
108. L 60001
109. L 60002
110. L 60003
111. N OBJ
112. COLUMNS

114.	C1AC1	UTIL1	-1.0	60001	1.0
115.	C1AC1	60002	2.0	60003	3.0
116.	C1AC2	UTIL1	-1.0	60001	2.0
117.	C1AC2	60002	2.0	60003	2.0
118.	C2AC1	UTIL2	-1.0	60001	4.0
119.	C2AC1	60002	3.	60003	1.
120.	C2AC2	UTIL2	-1.0	60001	3.
121.	C2AC2	60002	2.	60003	2.
122.	C3AC1	UTIL3	-1.	60002	1.0
123.	C3AC2	UTIL3	-1.	60001	1.0
124.	C3AC2	60002	2.		
125.	C3AC3	UTIL3	-1.	60002	3.
126.	C3AC3	60003	4.		
127.	C3AC4	UTIL3	-1.0	60001	3.
128.	C3AC4	60003	3.0		
129.	C4AC1	UTIL4	-1.0	60001	1.
130.	C4AC1	60002	1.0		
131.	C4AC2	UTIL4	-1.	60001	2.
132.	MEXP	60001	1.	60002	1.
133.	MEXP	60003	1.	OBJ	1.

RHS

135.	ENDOW	UTIL1	-0.48	UTIL2	-0.32
136.	ENDOW	UTIL3	-5.1	UTIL4	-0.95
137.	ENDOW	60001	5.	60002	11.0
138.	ENDOW	60003	9.		

ENDATA

140.	0	1	0		
141.	-1	5	7		
142.	2.		2.	1.	
143.					
144.	0	2	0		
145.	-1	5	7		
146.	1.		3.	3.	
147.					
148.	0	3			
149.	-1	5	7		
150.	1.		5.	4.	
151.					
152.	0	4			
153.	-1	5	7		
154.	1.		1.	1.	
155.					

```

J,IPRUD= 0,STPMX= 10.000000000000000 ,STPRD= .5000000000000000 ,ICFCHO= 1, IH= 4
0,KOUTR= 0,ITLIME= 100
&END
IECHO,IH,ICEFC 0 4 1
&PARAM
ZICLZE= .100000005E-03,ZIOLPV= .100000034E-05,ZICOST= .10000005E-03,ZSCALE= -1.00000000 ,NRMAX= 5,IFP
2000,
NEMAX= 4000,IALG= 0,NDEGI= 1,NDUAL= 999,NIPIW= 99999,KINP= 99999,7IOLRP= .99999975E-05,7
0,IFCRSH= 1,IOBJ= 8,ITCH= 30,INVFFQ= 30,ITPLIM= 99999,7IOLRP= .99999975E-05,7
00016E-01,
ZIFTA= .100000043E-11,KINB= 0,ZICLDA= .100000008E-08
&END
PROBLEM WHISMAN

```

```

.L=
350,NTMAX=
AS=
TC.RI= .100

```

```

PROBLEM STATISTICS
  8 ROWS
    11 STRUCTURAL COLUMNS
    36 NON-ZERO ELEMENTS
    DENSITY = 0.40909
    THE TIME LEFT IS NOW 1.00000 SEC.
    INVERT STATISTICS
      20 NONZ IN HASTS
      5 STRUCTURAL COLUMNS IN EASIS
      2 VECTORS ARCVB BUMP
      6 VECTORS HELCW BUMP
      8 NONZ 2 ETAS
      9 NONZ 3 ETAS
    TOTALS: 12 OFF DIAG NONZ 5 ETAS
    40: RELATIVE ERROR IN X = 0.0

```

ITCOUNT	STATUS	NINF	CRJ	VALUE	VEGIN	VECOUNT	NNEGDU	DJ	NFTA	NELLEM
1	I	2	9	15959899	C3AC3	MEXP	5	-6	5	17
2	I	1	0	58599997	C3AC4	GOOD2	4	-2	6	17
3	I	0	0	58874998	MEXP	GOOD1	1	50000000	7	22
4	F	0	-1	54874992	C2AC2	GOOD1	3	-1	8	26
5	F	0	-1	60599966	GOOD3	C3AC4	2	-1	9	31
5	F	0	-2	60599966	GOOD3	C3AC4	0	-0	9	37
5	F	0	-2	60599966	GOOD3	C3AC4	0	-0	10	42

JH(I)	VALUE	ROW NAMES	PI(I)	RHS
C1AC1	0.48000000	UTIL1	1.00000000	-0.48000000
C2AC2	0.32000000	UTIL2	3.00000000	-0.32000000
C3AC1	4.73000000	UTIL3	0.0	-5.10000000
C4AC1	0.95000000	UTIL4	1.00000000	-0.95000000
WEXP	2.61000000	GOOD1	1.00000000	5.00000000
GOOD3	2.83000000	GOOD2	0.0	11.00000000
C3AC3	0.37000000	GOOD3	0.0	9.00000000
HJ	-2.61000000	OBJ	-1.00000000	0.0

STRUCTURAL COLUMNS OF MATRIX.

558.
559.
560.
561.
562.
563.
564.
565.
566.
567.
568.
569.
570.
571.
572.
573.
574.
575.
576.
577.
578.
579.
580.
581.
582.
583.
584.
585.
586.
587.
588.
589.
590.
591.
592.
593.
594.
595.
596.
597.
598.
599.
600.
601.
602.
603.
604.
605.
606.
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.

```

COLUMNS 1 THROUGH 4
-0.4800 0.0 0.0 0.0
0.0 -0.3200 0.0 0.0
0.0 0.0 -5.1000 0.0
0.0 0.0 0.0 -0.9500
2.0000 1.0000 1.0000 1.0000
2.0000 3.0000 5.0000 1.0000
1.0000 3.0000 4.0000 1.0000
0.0 0.0 0.0 0.0

-----
THE 1TH ENDPT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
A3
STEPLNGTH= 0.0 0.0 0.0 0.0
ITERATION 0 X= 0.740000 CONSTRAINT TYPE:( 1, 7)
NORM(F(X))= 0.0 F= 0.0
AFTER 0 NEWTON ITERATIONS.
THE 15TH PRIMAL VARIABLE WENT TO ZERO.
VEGIN: UTIL1

-----
THE 2TH ENDPT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
A3
STEPLNGTH= 0.0 0.0 0.0 0.0
ITERATION 0 X= 0.500000 CONSTRAINT TYPE:( -1, 11)
NORM(F(X))= 0.0 F= 0.0
AFTER 0 NEWTON ITERATIONS.
THE 18TH DUAL VARIABLE WENT TO ZERO.
VEGIN: C4AC2
VEOUT: UTIL1

-----
THE 3TH ENDPT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
A3
STEPLNGTH= 0.0 0.0 0.0 0.0
ITERATION 0 X= 0.113333 CONSTRAINT TYPE:( 0, 9)
NORM(F(X))= 0.0 F= 0.0

SUPERBASIC INFO: KNU= 1 KNU= 0
NUH(I)= 1 0 0 0
MUH(I)= 0 0 0 0
T= 0.853333
LAMBDA= 1.500000 0.0 0.0 0.0
THE TIME LEFT IS NOW 2.500000 0.500000 0.0 0.0
AFTER 0 NEWTON ITERATIONS.

THE BILINEAR CONSTRAINT HIT ZERO.
THERE ARE NOW 1 BUDGET SUPPLUSES AT ZERO.

```

```

618. -----
619. THE 4TH ENDPT CALL:
620. THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
621. A3
622. 0.0 0.8533334
623. 1.3000000 0.0
624. STEPLENGTH= 0.113333 CCNSTRANT TYPE:( 1. 7)
625. ITERATION 0 X= 0.85333 0.11333
626. NORM(F(X))= 0.0 F= 0.0 0.0
627. AFTER 0 NEWTON ITERATIONS,
628. THE 13TH PRIMAL VARIABLE WENT TO ZERO.
629. VECIN: UTIL1
630. -----
631. THE 5TH ENDPT CALL:
632. THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
633. A3
634. -0.7474053 1.5000000
635. 0.6643638 0.1133334
636. 0.533091 CCNSTRANT TYPE:( 1. 5)
637. STEPLENGTH= 1.10156 0.46750
638. ITERATION 0 X= 0.141113 F= 0.14113 0.000000
639. NORM(F(X))= 1.18519 0.46750
640. ITERATION 1 X= 0.000000 F= 0.000000 0.0
641. NORM(F(X))=
642. THE QUADRATIC PICKED THE WRNG CCNSTRANT.
643. THE MOST INFREASIBLE CCNSTRANT IS TYPE 0 NUMBER 5
644. WITH A VALUE OF -0.846256
645. ITERATION 0 X= 1.62037 -0.24708
646. NORM(F(X))= 2.492337 F= 0.228929 2.421801
647. ITERATION 1 X= 1.13853 0.24310
648. NORM(F(X))= 0.472660 F= 0.334221 -0.334221
649. ITERATION 2 X= 1.32670 0.25484
650. NORM(F(X))= 0.013770 F= -0.009737 0.009737
651. ITERATION 3 X= 1.32152 0.29342
652. NORM(F(X))= 0.000010 F= -0.000007 0.000007
653. ITERATION 4 X= 1.32151 0.29342
654. NORM(F(X))= 0.000000 F= -0.000000 0.000000
655. AFTER 5 NEWTON ITERATIONS,
656. THE BILINEAR CCNSTRANT HIT ZERO.
657. THERE ARE NOW 2 BUDGET SURPLUSES AT ZERO.
658. -----
659. THE 6TH ENDPT CALL:
660. THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
661. A3
662. 0.3937402 1.3215135
663. 0.3841748 0.2934162
664. 0.453310 CCNSTRANT TYPE:( -1. 11)
665. STEPLENGTH= 1.50000 0.46757 0.37856
666. ITERATION 0 X= 0.047529 F= 0.036484 0.331083 0.000000
667. NORM(F(X))= 1.50000 0.46000 0.26667 0.0
668. ITERATION 1 X= 0.0 F= 0.0 0.0 0.0
669. NORM(F(X))=
670. SUPPRBASIC INFO: KNU= 2 KNU= 1
671. NUH(I)= 0 1 2 0
672. MUH(I)= 1 0 0 0
673.
674.
675.
676.
677.

```

```

678.      T = 0.853333      0.480000      0.366667      0.0
679.      LAMBDA = 1.500000      2.500000      0.500000      1.000000
680.      THE TIME LEFT IS NOW 1.000000 SEC.
681.      AFTER 1 NEWTON ITERATIONS,
682.      THE 18TH DUAL VARIABLE WENT TO ZERO.
683.      VECIN: C4AC2      VECOUT: UTIL1
684.
685. -----
686.      THE 7TH ENDPT CALL:
687.      THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
688.      A2(*U)+
689.      0.0      0.480000
690.      1.000000      0.366666
691.      0.0      0.853333
692.      STEPLENGTH= 0.533334      CONSTRAINT TYPE:( 0, 9)
693.      ITERATION 0 X= 0.480000      0.900000      0.853333      0.000000
694.      NORM(F(X))= 0.000000      F= -0.000000      0.0      0.000000
695.      AFTER 0 NEWTON ITERATIONS,
696.
697.      THE BILINEAR CONSTRAINT HIT ZERO.
698.      THERE ARE NOW 3 BUDGET SURPLUSES AT ZERO.
699.
700. -----
701.      THE 8TH ENDPT CALL:
702.      THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
703.      A2(*U)+
704.      0.0      0.480000
705.      0.0      0.500000
706.      0.0      0.853333
707.      1.000000      0.0
708.      STEPLENGTH= 0.050000      CONSTRAINT TYPE:( 1, 5)
709.      ITERATION 0 X= 0.480000      0.500000      0.853333      0.0
710.      NORM(F(X))= 0.000000      F= -0.000000      0.000000      0.0
711.      AFTER 0 NEWTON ITERATIONS,
712.
713.      AFTER 8 ENDPT CALLS, WE HAVE EQUILIBRIUM.
714.
715.      TOTALS:      SCALAR FUNCTION CALLS= 45
716.                JACOBIAN EVALUATIONS= 2
717.
718.      THE TIME LEFT IS NOW 1.000000 SEC.
719.      UTILITY LEVELS: 1.333334      0.800000      6.000000
720.      PRICES: 1.500000      2.500000      0.500000      1.000000
721.      WHISMAN
722.
723. -----
724.      JH(I)      VALUE      ROW NAMES      PI(I)      RHS
725.      C1AC1      1.33333337      UTIL1      1.50000000      -0.48000000
726.      C2AC2      0.80000001      UTIL2      2.50000000      -0.32000000
727.      C3AC1      6.00000057      UTIL3      0.50000000      -5.10000000
728.      C4AC1      0.733333267      UTIL4      1.00000000      -0.95000000
729.      NEXP      0.00000000      GOOD1      0.50000000      5.00000000
730.      GOOD3      3.39555986      GOOD2      0.50000000      11.00000000
731.      C4AC2      0.26666667      GOOD3      0.0      9.00000000
732.      CBJ      -0.00000000      OBJ      -1.00000000      0.0
733.
734. -----

```


II.5. Source Listing for BCA

The following pages are a listing of the program BCA as it is written in FORTRAN IV. The subroutines which make up LPM1 (Tomlin [1975]) are omitted.

```

1.      IMPLICIT REAL*8 (A-H,O-Z)
2.      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,RS,P,DD,DD1,RL1,RL2
3.      INTEGER*2 JH(350),DIGMA(9*2),KINBAS(1302),IDBAS(1302)
4.      INTEGER*2 ISTYPE,LA,LE,JA,IE,PUN,LC(20),IC(300)
5.      DOUBLE PRECISION F(8000)
6.      REAL A(4000),C(800),CMIN,COND,ERMAX,SUMINF
7.
8.      C
9.      COMMON DSUM,DPROD,CY,DE,DP,R(350),X(350),Y(350),YTEMP(350),
10.     1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
11.     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,(03J,IROWP,IVIN,IVOUT,
12.     3ITCNT,INVEEQ,ITRLIM,IFEEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
13.     4NGELEM,NINF,NUELEM,NUETA,NNEGDI,NLINES,ISTYPE(350),
14.     5LA(1302),LE(2002),PUN(8),
15.     6IPUNC,NDECI,NDUAL,NIPITW,IFEAS,IFCRSH
16.
17.      C
18.      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
19.      COMMON IA(4000),IF(8000)
20.      COMMON/LP1/P1(1302),XX(1302)
21.      COMMON/ALCST/BE1(10),BE2(10),E1(10),E2(10),C,IC,LC
22.      COMMON/ALCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
23.      COMMON/ALCNS/G1(350,10),G2(400,10),EA(350),PB(400)
24.      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
25.      COMMON/INDX2/ JH,DIGMA,KINEAS,IDEAS
26.      COMMON/SCALE/ BT,NH,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
27.      COMMON/UNIT/ IH,A,M,DD,KMU,KNU,RL1,RL2,MP1,NM,INO,INP
28.      COMMON/INT/ IPS(30),KDET,KCNT,ISING
29.      COMMON/TOLR/ TOLFZ,TOLRD,TOLCV,STPMX,STPRD
30.      DIMENSION VS(10),SHR(10)
31.      NAMEL(1)/PARM1/TOLFZ,TOLRD,TOLCV,ICNTRL,ICECHO,
32.      1STPMX,STPRD,ICECHO,IF,L,IPROD,KOUTB,ITLIME
33.
34.      C
35.      C
36.      C
37.      C
38.      C
39.      C
40.      C
41.      C
42.      C
43.      C
44.      C
45.      C
46.      C
47.      C
48.      C
49.      C
50.      C
51.      C
52.      C
53.      C
54.      C
55.      C
56.      C
57.      C
58.      C
59.      C
60.      C
61.      C
62.      C
63.      C
64.      C
65.      C
66.      C
67.      C
68.      C
69.      C
70.      C
71.      C
72.      C
73.      C
74.      C
75.      C
76.      C
77.      C
78.      C
79.      C
80.      C
81.      C
82.      C
83.      C
84.      C
85.      C
86.      C
87.      C
88.      C
89.      C
90.      C
91.      C
92.      C
93.      C
94.      C
95.      C
96.      C
97.      C
98.      C
99.      C
100.     C
101.     C
102.     C
103.     C
104.     C
105.     C
106.     C
107.     C
108.     C
109.     C
110.     C
111.     C
112.     C
113.     C
114.     C
115.     C
116.     C
117.     C
118.     C
119.     C
120.     C
121.     C
122.     C
123.     C
124.     C
125.     C
126.     C
127.     C
128.     C
129.     C
130.     C
131.     C
132.     C
133.     C
134.     C
135.     C
136.     C
137.     C
138.     C
139.     C
140.     C
141.     C
142.     C
143.     C
144.     C
145.     C
146.     C
147.     C
148.     C
149.     C
150.     C
151.     C
152.     C
153.     C
154.     C
155.     C
156.     C
157.     C
158.     C
159.     C
160.     C
161.     C
162.     C
163.     C
164.     C
165.     C
166.     C
167.     C
168.     C
169.     C
170.     C
171.     C
172.     C
173.     C
174.     C
175.     C
176.     C
177.     C
178.     C
179.     C
180.     C
181.     C
182.     C
183.     C
184.     C
185.     C
186.     C
187.     C
188.     C
189.     C
190.     C
191.     C
192.     C
193.     C
194.     C
195.     C
196.     C
197.     C
198.     C
199.     C
200.     C
201.     C
202.     C
203.     C
204.     C
205.     C
206.     C
207.     C
208.     C
209.     C
210.     C
211.     C
212.     C
213.     C
214.     C
215.     C
216.     C
217.     C
218.     C
219.     C
220.     C
221.     C
222.     C
223.     C
224.     C
225.     C
226.     C
227.     C
228.     C
229.     C
230.     C
231.     C
232.     C
233.     C
234.     C
235.     C
236.     C
237.     C
238.     C
239.     C
240.     C
241.     C
242.     C
243.     C
244.     C
245.     C
246.     C
247.     C
248.     C
249.     C
250.     C
251.     C
252.     C
253.     C
254.     C
255.     C
256.     C
257.     C
258.     C
259.     C
260.     C
261.     C
262.     C
263.     C
264.     C
265.     C
266.     C
267.     C
268.     C
269.     C
270.     C
271.     C
272.     C
273.     C
274.     C
275.     C
276.     C
277.     C
278.     C
279.     C
280.     C
281.     C
282.     C
283.     C
284.     C
285.     C
286.     C
287.     C
288.     C
289.     C
290.     C
291.     C
292.     C
293.     C
294.     C
295.     C
296.     C
297.     C
298.     C
299.     C
300.     C
301.     C
302.     C
303.     C
304.     C
305.     C
306.     C
307.     C
308.     C
309.     C
310.     C
311.     C
312.     C
313.     C
314.     C
315.     C
316.     C
317.     C
318.     C
319.     C
320.     C
321.     C
322.     C
323.     C
324.     C
325.     C
326.     C
327.     C
328.     C
329.     C
330.     C
331.     C
332.     C
333.     C
334.     C
335.     C
336.     C
337.     C
338.     C
339.     C
340.     C
341.     C
342.     C
343.     C
344.     C
345.     C
346.     C
347.     C
348.     C
349.     C
350.     C
351.     C
352.     C
353.     C
354.     C
355.     C
356.     C
357.     C
358.     C
359.     C
360.     C
361.     C
362.     C
363.     C
364.     C
365.     C
366.     C
367.     C
368.     C
369.     C
370.     C
371.     C
372.     C
373.     C
374.     C
375.     C
376.     C
377.     C
378.     C
379.     C
380.     C
381.     C
382.     C
383.     C
384.     C
385.     C
386.     C
387.     C
388.     C
389.     C
390.     C
391.     C
392.     C
393.     C
394.     C
395.     C
396.     C
397.     C
398.     C
399.     C
400.     C
401.     C
402.     C
403.     C
404.     C
405.     C
406.     C
407.     C
408.     C
409.     C
410.     C
411.     C
412.     C
413.     C
414.     C
415.     C
416.     C
417.     C
418.     C
419.     C
420.     C
421.     C
422.     C
423.     C
424.     C
425.     C
426.     C
427.     C
428.     C
429.     C
430.     C
431.     C
432.     C
433.     C
434.     C
435.     C
436.     C
437.     C
438.     C
439.     C
440.     C
441.     C
442.     C
443.     C
444.     C
445.     C
446.     C
447.     C
448.     C
449.     C
450.     C
451.     C
452.     C
453.     C
454.     C
455.     C
456.     C
457.     C
458.     C
459.     C
460.     C
461.     C
462.     C
463.     C
464.     C
465.     C
466.     C
467.     C
468.     C
469.     C
470.     C
471.     C
472.     C
473.     C
474.     C
475.     C
476.     C
477.     C
478.     C
479.     C
480.     C
481.     C
482.     C
483.     C
484.     C
485.     C
486.     C
487.     C
488.     C
489.     C
490.     C
491.     C
492.     C
493.     C
494.     C
495.     C
496.     C
497.     C
498.     C
499.     C
500.     C
501.     C
502.     C
503.     C
504.     C
505.     C
506.     C
507.     C
508.     C
509.     C
510.     C
511.     C
512.     C
513.     C
514.     C
515.     C
516.     C
517.     C
518.     C
519.     C
520.     C
521.     C
522.     C
523.     C
524.     C
525.     C
526.     C
527.     C
528.     C
529.     C
530.     C
531.     C
532.     C
533.     C
534.     C
535.     C
536.     C
537.     C
538.     C
539.     C
540.     C
541.     C
542.     C
543.     C
544.     C
545.     C
546.     C
547.     C
548.     C
549.     C
550.     C
551.     C
552.     C
553.     C
554.     C
555.     C
556.     C
557.     C
558.     C
559.     C
560.     C
561.     C
562.     C
563.     C
564.     C
565.     C
566.     C
567.     C
568.     C
569.     C
570.     C
571.     C
572.     C
573.     C
574.     C
575.     C
576.     C
577.     C
578.     C
579.     C
580.     C
581.     C
582.     C
583.     C
584.     C
585.     C
586.     C
587.     C
588.     C
589.     C
590.     C
591.     C
592.     C
593.     C
594.     C
595.     C
596.     C
597.     C
598.     C
599.     C
600.     C
601.     C
602.     C
603.     C
604.     C
605.     C
606.     C
607.     C
608.     C
609.     C
610.     C
611.     C
612.     C
613.     C
614.     C
615.     C
616.     C
617.     C
618.     C
619.     C
620.     C
621.     C
622.     C
623.     C
624.     C
625.     C
626.     C
627.     C
628.     C
629.     C
630.     C
631.     C
632.     C
633.     C
634.     C
635.     C
636.     C
637.     C
638.     C
639.     C
640.     C
641.     C
642.     C
643.     C
644.     C
645.     C
646.     C
647.     C
648.     C
649.     C
650.     C
651.     C
652.     C
653.     C
654.     C
655.     C
656.     C
657.     C
658.     C
659.     C
660.     C
661.     C
662.     C
663.     C
664.     C
665.     C
666.     C
667.     C
668.     C
669.     C
670.     C
671.     C
672.     C
673.     C
674.     C
675.     C
676.     C
677.     C
678.     C
679.     C
680.     C
681.     C
682.     C
683.     C
684.     C
685.     C
686.     C
687.     C
688.     C
689.     C
690.     C
691.     C
692.     C
693.     C
694.     C
695.     C
696.     C
697.     C
698.     C
699.     C
700.     C
701.     C
702.     C
703.     C
704.     C
705.     C
706.     C
707.     C
708.     C
709.     C
710.     C
711.     C
712.     C
713.     C
714.     C
715.     C
716.     C
717.     C
718.     C
719.     C
720.     C
721.     C
722.     C
723.     C
724.     C
725.     C
726.     C
727.     C
728.     C
729.     C
730.     C
731.     C
732.     C
733.     C
734.     C
735.     C
736.     C
737.     C
738.     C
739.     C
740.     C
741.     C
742.     C
743.     C
744.     C
745.     C
746.     C
747.     C
748.     C
749.     C
750.     C
751.     C
752.     C
753.     C
754.     C
755.     C
756.     C
757.     C
758.     C
759.     C
760.     C
761.     C
762.     C
763.     C
764.     C
765.     C
766.     C
767.     C
768.     C
769.     C
770.     C
771.     C
772.     C
773.     C
774.     C
775.     C
776.     C
777.     C
778.     C
779.     C
780.     C
781.     C
782.     C
783.     C
784.     C
785.     C
786.     C
787.     C
788.     C
789.     C
790.     C
791.     C
792.     C
793.     C
794.     C
795.     C
796.     C
797.     C
798.     C
799.     C
800.     C
801.     C
802.     C
803.     C
804.     C
805.     C
806.     C
807.     C
808.     C
809.     C
810.     C
811.     C
812.     C
813.     C
814.     C
815.     C
816.     C
817.     C
818.     C
819.     C
820.     C
821.     C
822.     C
823.     C
824.     C
825.     C
826.     C
827.     C
828.     C
829.     C
830.     C
831.     C
832.     C
833.     C
834.     C
835.     C
836.     C
837.     C
838.     C
839.     C
840.     C
841.     C
842.     C
843.     C
844.     C
845.     C
846.     C
847.     C
848.     C
849.     C
850.     C
851.     C
852.     C
853.     C
854.     C
855.     C
856.     C
857.     C
858.     C
859.     C
860.     C
861.     C
862.     C
863.     C
864.     C
865.     C
866.     C
867.     C
868.     C
869.     C
870.     C
871.     C
872.     C
873.     C
874.     C
875.     C
876.     C
877.     C
878.     C
879.     C
880.     C
881.     C
882.     C
883.     C
884.     C
885.     C
886.     C
887.     C
888.     C
889.     C
890.     C
891.     C
892.     C
893.     C
894.     C
895.     C
896.     C
897.     C
898.     C
899.     C
900.     C
901.     C
902.     C
903.     C
904.     C
905.     C
906.     C
907.     C
908.     C
909.     C
910.     C
911.     C
912.     C
913.     C
914.     C
915.     C
916.     C
917.     C
918.     C
919.     C
920.     C
921.     C
922.     C
923.     C
924.     C
925.     C
926.     C
927.     C
928.     C
929.     C
930.     C
931.     C
932.     C
933.     C
934.     C
935.     C
936.     C
937.     C
938.     C
939.     C
940.     C
941.     C
942.     C
943.     C
944.     C
945.     C
946.     C
947.     C
948.     C
949.     C
950.     C
951.     C
952.     C
953.     C
954.     C
955.     C
956.     C
957.     C
958.     C
959.     C
960.     C
961.     C
962.     C
963.     C
964.     C
965.     C
966.     C
967.     C
968.     C
969.     C
970.     C
971.     C
972.     C
973.     C
974.     C
975.     C
976.     C
977.     C
978.     C
979.     C
980.     C
981.     C
982.     C
983.     C
984.     C
985.     C
986.     C
987.     C
988.     C
989.     C
990.     C
991.     C
992.     C
993.     C
994.     C
995.     C
996.     C
997.     C
998.     C
999.     C
1000.    C

```

```

48.      IF (ICNTRL) 10,40,40
49.      10 CALL INPUT(IACT)
50.      IF (IACT) 30, 20, 20
51.      20 IF (IECHO.EQ.0) GO TO 29
52.      MUB= NROW
53.      WRITE (6,21)
54.      21 FORMAT(/////,'          STRUCTURAL COLUMNS OF MATRIX.')
```

22 LB= MUB+1
MUB= LB+ 14
K= 0
DO 25 J= LB,MUB
IF (J.GT.NCOL) GO TO 26
K= K+ 1
CALL UNPACK(J)
DO 24 I= 1,NROW
G1(I,K)= Y(I)

24 CONTINUE
25 CONTINUE
26 WRITE (6,27) LB,MUB
27 FORMAT (/////,' COLUMNS ',I4,' THROUGH ',I4)
DO 23 I=1,NROW
WRITE (6,28) (G1(I,J),J=1,K)
28 FORMAT (1X,15F8.3)
23 CONTINUE
IF (MUB .LT. NCOL) GO TO 22
29 CALL NORMAL
CALL UNRAVL(0)
GO TO 10
30 STOP

C 40 CALL INPUT(IACT)
CALL DEBUG(1)
CALL NORMAL
CALL UNRAVL(0)

C 41 CALL DEBUG(1)
CALL SHIFTR(3,4)
KEND= 0
KENDSV= 0
IHL= IH+ L
M= NROW
N= NCOL- NROW
NM= N+ M
LI= M- L
MMI= M- 1
ITSINV= 1
IF (IPROD.EQ.0) GO TO 43
READ (5,1030) (SHR(I),I=1,IH)

43 ICCU=0
ISAV= 0
LC(1)= 1
DO 68 K= 1,IH
KPI= K+ 1

45 READ(KINP,1010) ISFR, LL, KK
IF (ISFR.EQ.1) GO TO 50
IF (ISFR.EQ.-1) GO TO 42
IF (LL.EQ.0) GO TO 58
IF (KK.NE.0) GO TO 47
ICCU= ICCU+ 1
IC(ICCU)= LL


```

108.          C(ICOU)= B(LL)
109.          GO TO 45
110. 47      DO 48 I= LL, KK
111.          ICCU= ICCU+ 1
112.          IC(ICCU)= I
113.          C(ICCU)= B(I)
114. 48      CONTINUE
115.          GO TO 45
116. 42      IF (KK.NE.0) GO TO 44
117.          ICCU= ICCU+ 1
118.          IC(ICCU)= LL
119.          READ (5,1030) C(ICCU)
120.          GO TO 45
121. 44      ISAV= ICCU+ 1
122.          DO 46 I= LL, KK
123.              ICCU= ICCU+ 1
124.              IC(ICCU)= I
125. 46      CONTINUE
126.          READ (5,1030) (C(J),J=ISAV,ICCU)
127.          GO TO 45
128. 50      IF (KK.NE.0) GO TO 53
129.          ICCU= ICCU+ 1
130.          IC(ICCU)= LL
131.          C(ICCU)= SHR(K)*B(LL)
132.          GO TO 45
133. 53      DO 55 I= LL, KK
134.          ICCU= ICCU+ 1
135.          C(ICCU)= SHR(K)*B(I)
136.          IC(ICCU)= I
137. 55      CONTINUE
138.          GO TO 45
139. 58      LC(KP1)= ICCU+1
140. 68      CONTINUE
141.          IF (ICECHO.EQ.0) GO TO 69
142.          MUB= 0
143.          WRITE (6,61)
144. 61      FORMAT(/////,'          STRUCTURAL COLUMNS OF MATRIX.')
```

```

145. 62      LB= MUB+1
146.          MUB= LB+ IH - 1
147.          K= 0
148.          DO 65 J= LB, MUB
149.              IF (J.GT.NCOL) GO TO 66
150.              K= K+ 1
151.              CALL UPAKC(J)
152.              DO 64 I= 1, NROW
153.                  G1(I,K)= Y(I)
154. 64      CONTINUE
155. 65      CONTINUE
156.          WRITE (6,67) LB, MUB
157. 67      FORMAT (/////,' COLUMNS ',14,' THROUGH ',14)
158.          DO 74 I= 1, NROW
159.              WRITE (6,73) (G1(I,J),J=1,IH)
160. 73      FORMAT (1X,10F10.4)
161. 74      CONTINUE
162. 69      DO 70 I= 1, M
163.          IR= JH(I)
164.          XX(IR)= X(I)
165.          PI(I)= YTEMP(I)
166.          BA(I)= X(I)
167. 70      CONTINUE

```

```

168.      MPI= M+ 1
169.      DO 100 J= MPI,NM
170.      IF (KINBAS(J).NE. 0) GO TO 90
171.      DSUM= 0.
172.      LL= LA(J)
173.      KK= LA(J+1) - 1
174.      DO 80 I= LL,KK
175.      IR= IA(I)
176.      DE= A(I)
177.      DPRCD= DE*YTEMP(IR)
178.      DSUM= DSUM + DPRCD
179.      80 CONTINUE
180.      PI(J)= DSUM
181.      GO TO 100
182.      90 PI(J)= 0.
183.      100 CONTINUE
184.      K= 0
185.      DO 110 I= 1,NM
186.      IF (KINBAS(I) .EQ. 0) GO TO 105
187.      IDBAS(I)= 0
188.      GO TO 110
189.      105 K= K+ 1
190.      DIGMA(K)= I
191.      BB(K)= PI(I)
192.      IDBAS(I)= K
193.      XX(I)= 0.0
194.      110 CONTINUE
195.      C      THIS IS THE BILINEAR PHASE OF THE ALGORITHM. VARIABLES XX(1).....
196.      C      XX(M) ARE THE SLACKS. AND PI(1).....PI(M) ARE THE USUAL PI'S.
197.      C      XX(M+1).....XX(M+N) ARE THE X'S. PI(M+1).....PI(M+N) ARE THE
198.      C      DUAL SLACKS. MORE INITIALIZATIONS.
199.      C
200.      DO 120 I=1,IH
201.      MUH(I)= 0
202.      NUH(I)= 0
203.      NU(I)= 0
204.      MU(I)= 0
205.      120 CONTINUE
206.      KNU= 0
207.      KMU= 0
208.      DD=1
209.      420 PD= 1
210.      IF (KINBAS(DD).GT.0) GO TO 425
211.      CALL SUPERB(1,1,DD,0,0)
212.      GO TO 430
213.      425 CALL SUPERB(1,-1,DD,0,0)
214.      PD= -1
215.      430 JS= DD
216.      CALL BLCNS
217.      ZFLAG= 0
218.      GO TO 460
219.      450 NTEMP(1)= NTEMP(1)+ JTINV
220.      CALL INVERT
221.      CALL RECALC
222.      ITSINV= 0
223.      460 KEND= KEND+ 1
224.      ITSINV= ITSINV + 1
225.      IF (KEND .GT. ITIME) STOP
226.      WRITE (6,222) KEND
227.      222 FORMAT (/, '-----

```

```

223.      1,7,' THE ',14,'TH ENDENT CALL:')
229.      CALL ENDPNT(JS,PD1,IS,NET)
230.      IF ( (KEND/3)*3 .NE. KEND ) GO TO 468
231.      CALL DEBUG(4)
232.      CALL DEBUG(1)
233.      468 WRITE (6,470) NET
234.      470 FORMAT (1X,' AFTER ',13,' NEWTON ITERATIONS,')
235.      IF (PD1.EQ.1 .AND. IS.EQ.NN) GO TO 920
236.      IF (PD1) 300,305,210
237.      300 WRITE (6,305) IS
238.      305 FORMAT (1X,' THE ',13,'TH DUAL VARIABLE WENT TO ZERO.')
```

GO TO 480

```

239.      306 WRITE (6,307) DD
240.      307 FORMAT (7,1X,' THE BILINEAR CONSTRAINT HIT ZERO.')
```

1, ' THERE ARE NOW ',13,' BUDGET SURPLUSES AT ZERO.')

```

242.      IF (DD.EQ.1H) GO TO 920
243.      DD= DD+1
244.      GO TO 420
245.      310 WRITE (6,315) IS
246.      315 FORMAT (1X,' THE ',13,'TH PRIMAL VARIABLE WENT TO ZERO.')
```

480 IF (PD1.EQ.-1) GO TO 520

```

248.      IF (IS.EQ. DD) GO TO 490
249.      C
250.      C      A PRIMAL VARIABLE WENT TO ZERO. THAT VARIABLE MUST GO TO
251.      C      NONBASIC, AND THE INCOMING VARIABLE IS DETERMINED BY FINDING THE
252.      C      LARGEST ELEMENT IN THE APPROPRIATE ROW OF G1.
253.      C
254.      PD= -1
255.      483 IROWP= KINBAS(IS)
256.      CALL FINDER(1,IS,JCOLP)
257.      INNAM1= ICNAM(JCOLP,1)
258.      INNAM2= ICNAM(JCOLP,2)
259.      IONAM1= ICNAM(IS,1)
260.      IONAM2= ICNAM(IS,2)
261.      WRITE(6,485) INNAM1,INNAM2,IONAM1,IONAM2
262.      485 FORMAT(1X,' VECIN: ',2A4,' VECOUT: ',2A4)
263.      CALL PIVOT(JCOLP,IS)
264.      CALL UNPACK(JCOLP)
265.      CALL FTEAM(1)
266.      CALL WRTA
267.      CALL BSCNC(JCOLP,IS)
268.      CALL SUPERB(0,-1,JCOLP,1,JCOLP)
269.      CALL BLCGNS
270.      JS= IS
271.      IF (ITSINV .GE. INVERG) GO TO 450
272.      GO TO 460
273.      490 IDS= DD
274.      DD= DD -1
275.      JS= DD
276.      IF (DD.EQ.0) STOP
277.      PD= 0
278.      IF (KINBAS(IDS) .NE. 0) GO TO 500
279.      CALL SUPERB(2,0,0,1,IDS)
280.      GO TO 460
281.      C
282.      500 CALL SUPERB(2,0,0,-1,IDS)
283.      GO TO 460
284.      C
285.      C      A DUAL VARIABLE WENT TO ZERO. THAT VARIABLE MUST ENTER THE
286.      C      BASIS, AND THE LEAVING VARIABLE MUST BE DETERMINED BY FINDING
287.      C
```



```

288. C THE LARGEST PIVOT ELEMENT IN THE IOBAS(IS)-ROW OF G2.
289. C
290. 520 PD= 1
291. IF (IS.EQ.DD) GO TO 530
292. 523 JCCLP= IS
293. CALL FINDP(-1,IS,P)
294. IROWP= KINEAS(P)
295. INNAM1= ICNAM(IS,1)
296. INNAM2= ICNAM(IS,2)
297. IONAM1= ICNAM(P,1)
298. IONAM2= ICNAM(P,2)
299. WRITE (6,485) INNAM1,INNAM2,IONAM1,IONAM2
300. CALL PIVOT(JCCLP,P)
301. CALL UNPACK(JCCLP)
302. CALL FIPAN(1)
303. CALL WRETA
304. CALL HSCN(JCCLP,P)
305. CALL SUPERP(0,1,P,-1,P)
306. CALL BLCONS
307. JS= IS
308. IF (ITSINV.GE.INVERO) GO TO 450
309. GO TO 460
310. 530 PD= 0
311. IDS= DD
312. DD= DD- 1
313. JS= DD
314. IF (DD.LE.0) STOP
315. IF (ICPAS(IDS).NE. 0) GO TO 540
316. CALL SUPERP(2,0,0,-1,IDS)
317. GO TO 460
318. C
319. 540 CALL SUPERP(2,0,0,1,IDS)
320. GO TO 523
321. C
322. C
323. 920 WRITE (6,930) KEND
324. 930 FORMAT(/, ' AFTER ', I4, ' ENDPOINT CALLS, WE HAVE EQUILIBRIUM. ')
325. KJAC= KJAC/ (IH*IH)
326. WRITE (6,920) KEND, KJAC
327. 320 FORMAT(/, ' TOTALS: SCALAR FUNCTION CALLS= ', I3, /, ' IIX, ' JACOBIAN EVA
328. 1LLATIONS= ', I8, /)
329. CALL DEBUC(1)
330. DO 935 I= 1, IH
331. 935 VS(I)= -B(I)+ XX(I)
332. WRITE (6,940) (VS(I), I= 1, IH)
333. WRITE (6,945) (PI(I), I= 1, IH)
334. 940 FORMAT (1X, ' UTILITY LEVELS: ', 6F15.7)
335. 945 FORMAT (1X, ' PRICES: ', 9F12.6)
336. DO 950 I= 1, NROW
337. IV= JH(I)
338. X(I)= XX(IV)
339. Y(I)= PI(I)
340. 950 CONTINUE
341. CALL UNRAVL(0)
342. STOP
343. 1010 FORMAT (15I4)
344. 1030 FORMAT (10F7.4)
345. END
346. SUBROUTINE BLCONS
347. C

```

```

348. C WE ARE GOING TO CALCULATE THE DD POWS OF COEFFICIENTS
349. C FOR THE BILINEAR EQUATIONS AND THE BILINEAR INEQUALITY.
350. C THE BASIC MATHEMATICAL STRUCTURE IS THE FOLLOWING:
351. C
352. C      BF1 + D1*PI(MU) - DIAG(PI(MU))*(F1*X(NU) + E1) = 0
353. C      BF2 + D2*PI(MU) - DIAG(X(NU))*(F2*PI(MU) + E2) = 0
354. C
355. C      IMPLICIT REAL*8 (A-H,O-Z)
356. C      INTEGER PD,PD1,PD2,SS,FF,7FLAG,RS,P,DD,DD1,BL1,BL2
357. C      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
358. C      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
359. C      DOUBLE PRECISION F(800)
360. C      REAL A(4000),C(800),CMIN,COND,ERMAX,SUMINF
361. C
362. C      COMMON DSUM,DPROG,DY,DE,DP,F(350),X(350),Y(350),YTEMP(350),
363. C      IA,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
364. C      2NTEMP(20),KINP,ITIM,ITINV,JTINV,MSTAR,IOBJ,IRGWP,IVIN,IVOUT,
365. C      3ITCNT,INVERO,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NLEEM,NETA,NLELEM,NLETA,
366. C      4NGELEM,NINF,NUELEM,NUETA,NNECDJ,NLINES,ISTYPE(350),
367. C      5LA(1302),IF(2002),PUN(F),
368. C      6IPUNC,NDEGI,NDUAL,NIPIN,IFEAS,IFCRSH
369. C      COMMON IICH,ITCHA,IFRIWT,IFNEG,KOUTR
370. C      COMMON IA(4000),IE(8000)
371. C      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
372. C      COMMON/LP1/PI(1302),XX(1302)
373. C      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
374. C      COMMON/LNCCNS/G1(350,10),G2(400,10),BA(350),B3(400)
375. C      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
376. C      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
377. C      COMMON/SCAL/ BT,NB,JJ,VFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
378. C      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MD1,NM,INQ,INP
379. C      K1= 0
380. C      K2= 0
381. C      BL1= 0
382. C      BL2= 0
383. C      INQ= NUH(DD)
384. C      INP= NUH(CD)
385. C      IF (KMU.EQ.0) GO TO 100
386. C      DO 80 K= 1,KMU
387. C      K1= MU(K)
388. C      IF (K1.GT.IH) GO TO 80
389. C      IDD= KINBAS(K1)
390. C      BL1= BL1+ 1
391. C      IF (KNU.EQ.0) GO TO 45
392. C      DO 40 I= 1,KNU
393. C      40 F1(K,I)= G1(IDD,I)
394. C      45 E1(K)= BA(IDD)
395. C      CALL UPAKC(K1)
396. C      DO 60 I= 1,KMU
397. C      DSUM = 0.0
398. C      IF (MU(I).LE.M) DSUM= Y(MU(I))
399. C      DO 50 J= 1,M
400. C      IDJ= IDEAS(J)
401. C      IF (IDJ.EQ.C) GO TO 50
402. C      DSUM= DSUM+ Y(J)*G2(IDJ,I)
403. C      50 CONTINUE
404. C      DI(K,I)= DSUM
405. C      60 CONTINUE
406. C      BF1(K)= 0.
407. C      DO 70 J= 1,M

```

```

408.      IF (IDBAS(J).EQ.0) GO TO 70
409.      BF1(K)= BF1(K)+ Y(J)*BE(IDBAS(J))
410. 70 CONTINUE
411.      IF (K1.NE.DD) GO TO 80
412.      INEG= 1
413.      BL1= BL1-1
414. 80 CONTINUE
415. C
416. C      NOW THE SECOND TYPE OF EQUATION IS CALCULATED BECAUSE
417. C      PI(K) IS BASIC.
418. C
419. 100 IF (KNU.EQ.0) RETURN
420.      DD 145 K= 1,KNU
421.      K2= NU(K)
422.      IF (K2.CT.1F) GO TO 145
423.      IDD= IDBAS(K2)
424.      BL2= BL2+ 1
425.      IF (KMU.EQ.0) GO TO 123
426.      DD 120 I= 1,KMU
427. 120 F2(K,I)= G2(IDD,I)
428. 123 E2(K)= BE(IDD)
429.      CALL UPAKC(K2)
430.      IF (KMU.EQ.0) GO TO 132
431.      DD 131 I= 1,KMU
432.      D2(K,I)= 0.0
433.      IF (MU(I).LE. M) D2(K,I)= Y(MU(I))
434.      DD 125 J= 1,M
435.      IF (IDBAS(J).EQ.0) GO TO 125
436.      D2(K,I)= D2(K,I)+ Y(J)*G2(IDBAS(J),I)
437. 125 CONTINUE
438. 130 CONTINUE
439. 132 BF2(K)= 0.
440.      DD 140 J= 1,M
441.      IF (IDBAS(J).EQ.0) GO TO 140
442.      BF2(K)= BF2(K)+ Y(J)*BR(IDBAS(J))
443. 140 CONTINUE
444.      IF (K2.NE.DD) GO TO 145
445.      INEG= 2
446.      BL2= BL2- 1
447. 145 CONTINUE
448.      RETURN
449.      END
450.      SUBROUTINE FINDP(PD1,IS,P)
451. C
452. C      THIS SUBROUTINE CHOOSES THE VARIABLE TO BECOME IMPLICITLY
453. C      BASIC AS THE ONE WITH THE PIVOT ELEMENT LARGEST IN
454. C      ABSOLUTE VALUE
455. C
456.      IMPLICIT REAL*8 (A-H,O-Z)
457.      INTEGER PD,PD1,PD2,SS,FR,ZFLAG,RS,P,DD,DD1,BL1,BL2
458.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
459.      COMMON/LNCONS/G1(350,10),G2(400,10),EA(350),BB(400)
460.      COMMON/INDX1/NUF(10),MUH(10),NU(10),MU(10)
461.      COMMON/INDX2/JH,DIGMA,KINBAS,IDBAS
462.      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INO,IND
463.      COMMON/TOLER/ TOLF7,TCLPD,TCLCV,STEMX,STORD
464.      P= 0
465.      IF (PD1.EQ.-1) GO TO 30
466.      IDD= KINBAS(15)
467.      DD 20 I= 1,KNU

```



```

468.      IF (P.NE.0) GO TO 10
469.      BIG= DABS(G1(IDO,1))
470.      P= NU(1)
471.      GO TO 20
472. 10 COMP= DABS(G1(IDO,1))
473.      IF (COMP.LE.BIG) GO TO 20
474.      BIG= COMP
475.      P= NU(1)
476. 20 CONTINUE
477.      IF (BIG.LT.TOLFZ) P= 0
478.      RETURN
479. 3) IDO= IDBAS(15)
480.      DO 40 J= 1,KMU
481.      IF (P.NE.0) GO TO 35
482.      BIG= DABS(G2(IDO,J))
483.      P= MU(J)
484. 35 COMP= DABS(G2(IDO,J))
485.      IF (COMP.LE.BIG) GO TO 40
486.      BIG= COMP
487.      P= MU(J)
488. 40 CONTINUE
489.      IF (BIG.LT.TOLFZ) P= 0
490.      RETURN
491.      END
492.      SUBROUTINE PIVOT (SS,RR)
493.  C
494.  C      THIS ROUTINE PERFORMS A PIVOT ON THE PRIMAL SUPER-
495.  C      BASIC COLUMNS IF PD.EQ.1. DUAL IF PD.EQ.-1. THE PIVOT
496.  C      BRINGS COLUMN SS INTO THE BASIS AND COLUMN RR OUT OF THE
497.  C      BASIS.
498.  C
499.      IMPLICIT REAL*8 (A-H,O-Z)
500.      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,RS,P,DD,DD1,RL1,RL2
501.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
502.      INTEGER*2 ISTYPE,LA,LE,IA,IF,PUN,LC(20),IC(800)
503.      DOUBLE PRECISION E(8000)
504.      REAL A(4000),C(800),CMIN,COND,ERMAX,SUMINF
505.  C
506.      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),ZB(350),
507.      1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
508.      2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,I09J,IROWP,IVIN,IVOUT,
509.      3ITCNT,INVERO,ITRLIN,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
510.      4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
511.      5LA(1302),LE(2002),PUN(8),
512.      6IPUNC,NDECI,NDUAL,NIPIW,IFBAS,IFCRSH
513.      COMMON ITCB,ITCHA,IFPIW1,IFNEG,KOUTR
514.      COMMON IA(4000),IE(8000)
515.      COMMON/LP1/PI(1302),XX(1302)
516.      COMMON/LNCONS/G1(350,10),G2(400,10),FA(350),PB(400)
517.      COMMON/INDX1/ NUH(10),NUH(10),NU(10),MU(10)
518.      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
519.      COMMON/DIV/ IH,N,M,DD,KMU,KNU,RL1,RL2,MP1,NM,INO,IND
520.      EPS= 16.**(-13)
521.      RB= KINBAS(RR)
522.      IF (SS.GT. IH) GO TO 5
523.      IF (NUH(SS).NE.0) GO TO 20
524. 5 CALL UNPACK(SS)
525.      CALL FTRAN(1)
526.      Z( RB)= -1./Y(RB)
527.      DO 10 I= 1,M

```

```

528.      IF (I.NE.RB) ZB(I)= Y(I)*ZB(RB)
529.      10 CONTINUE
530.      GO TO 30
531.      20 JS= NUF(SS)
532.      ZB(RB)= 1./G1(RB,JS)
533.      DO 25 I= 1,M
534.      IF (I.NE.RB) ZB(I)= -C1(I,JS)*ZB(RB)
535.      25 CONTINUE
536.      30 IF (KNU.EQ.0) GO TO 55
537.      DO 30 J= 1,KNU
538.      IF (NU(J).EQ.SS) GO TO 50
539.      V= G1(RB,J)
540.      G1(RB,J)= 0.
541.      DO 40 I= 1,M
542.      G1(I,J)= C1(I,J)+ V*ZB(I)
543.      G1(RB,J)= -G1(RB,J)
544.      50 CONTINUE
545.      V= BA(RB)
546.      BA(RB)= 0.
547.      DO 60 I= 1,M
548.      BA(I)= BA(I)+ V*ZB(I)
549.      BA(RB)= -BA(RB)
550.      C
551.      C      WE ARE NOW PIVOTING ON THE DUAL SYSTEM. IF THIS
552.      C      IS NOT AN IMPLICIT EASIC-TYPE PIVOT, WE CALL SUPERB
553.      C      TO CALCULATE THE PIVOT COLUMN.
554.      C
555.      IFL= 0
556.      RH= IDBAS(SS)
557.      IF (RR.GT. IH) GO TO 70
558.      IF (MUF(RR).NE.0) GO TO 80
559.      70 IFL= 1
560.      CALL SUPERB(1,-1,RR,0.0)
561.      JS= KNU
562.      GO TO 82
563.      80 JS= MUF(RR)
564.      82 ZB(RB)= 1./G2(RB,JS)
565.      DO 85 I= 1,N
566.      IF (I.NE.RB) ZB(I)= -G2(I,JS)*ZB(RB)
567.      85 CONTINUE
568.      IF (KMU.EQ.0) GO TO 115
569.      DO 110 J= 1,KMU
570.      IF (MU(J).EQ.RR) GO TO 110
571.      V= G2(RB,J)
572.      G2(RB,J)= 0.
573.      DO 100 I= 1,N
574.      G2(I,J)= G2(I,J)+ V*ZB(I)
575.      G2(RB,J)= -G2(RB,J)
576.      100 CONTINUE
577.      115 V= EB(RB)
578.      BB(RB)= 0.0
579.      DO 120 I= 1,N
580.      BB(I)= EB(I)+ V*ZB(I)
581.      BB(RB)= -EB(RB)
582.      IF (IFL.EQ.0) RETURN
583.      CALL SUPERB(2,0,0,-1,RR)
584.      RETURN
585.      END
586.      C
587.      C

```

```

588.      SUBROUTINE UPACK(I1)
589.      IMPLICIT REAL*8 (A-H,O-Z)
590.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
591.      INTEGER*2 ISTYPE,LA,LE,IA,IF,PUN,LC(20),IC(400)
592.      DOUBLE PRECISION E(8000)
593.      REAL A(4000),C(800),CMIN,COND,ERMAX,SUMINF
594.      C
595.      COMMON DSUM,DPROD,DY,DE,DF,B(350),X(350),Y(350),YTEMP(350),
596.      1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
597.      2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,I03J,IROWP,IVIN,IVOUT,
598.      3ITCNT,INVERQ,ITRLIM,IFFE2,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
599.      4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
600.      5LA(1302),LE(2002),PUN(E),
601.      6IPUNC,NDFGI,NDUAL,NIPW,IFEAS,IFCRSH
602.      COMMON ITC,ITCHA,IFPIWT,IFNEG,KOUTP
603.      COMMON IA(4000),IF(9000)
604.      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
605.      C
606.      DO 100 I= 1,NROW
607.      Y(I)= 0.
608.      100 CONTINUE
609.      C
610.      LL= LC(I1)
611.      KK= LC(I1+1) - 1
612.      DO 200 I= LL,KK
613.      IR= IC(I)
614.      Y(IR)= C(I)
615.      200 CONTINUE
616.      C
617.      RETURN
618.      END
619.      C
620.      C THE FOLLOWING ROUTINE MAKES THE CHANGES IN THE INDEX
621.      C SETS NECESSARY EVERY TIME A BASIS CHANGE IS MADE.
622.      C
623.      SUBROUTINE BCCNC(S,R)
624.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
625.      COMMON/INX2/JH,DIGMA,KINBAS,IDBAS
626.      INTEGER S,SSAV,R,RSAB
627.      RSAB= KINBAS(R)
628.      JH(RSAB)= S
629.      KINBAS(R)= 0
630.      KINBAS(S)= RSAB
631.      SSAV= IDBAS(S)
632.      DIGMA(SSAV)= R
633.      IDBAS(S)= 0
634.      IDBAS(R)= SSAV
635.      RETURN
636.      END
637.      C THIS SUBROUTINE CAN DO THREE THINGS DETERMINED BY THE
638.      C PARAMETER 'KEY'. IT CAN ADD A COLUMN, REMOVE A COLUMN, OR
639.      C BOTH ADD AND REMOVE COLUMNS FROM THE PRIMAL OR DUAL SUPER-
640.      C BASIC COLUMNS, DEPENDING ON WHETHER KEY IS 1, 2, OR 0, RESP-
641.      C ECTIVELY. PD1 IS 1 OR -1 DEPENDING ON WHETHER A PRIMAL OR
642.      C DUAL COLUMN IS BEING ADDED. PD2 IS A SIMILAR FLAG FOR THE
643.      C COLUMN BEING REMOVED. IS AND JS INDICATE THE PARTICULAR
644.      C COLUMN TO BE ADDED OR REMOVED FROM THE GROUP OF COLUMNS
645.      C SPECIFIED BY PD1 AND PD2.
646.      C
647.      SUBROUTINE SUPERE(KEY,PD1,IS,PD2,JS)

```



```

648.      IMPLICIT REAL*8 (A-H,O-Z)
649.      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,RS,P,DD,DD1,BL1,BL2
650.      INTEGER*2 JH(350),DIGMA(952),KINPAS(1302),IDBAS(1302)
651.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(80)
652.      DOUBLE PRECISION E(8000)
653.      REAL A(4000),C(800),CMIN,COND,ERMAX,SUMINF
654.
655.      C
656.      COMMON DSLM,DPRDD,DY,DE,DF,R(350),X(350),Y(350),YTEMP(350),
657.      1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
658.      2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,I09J,IROWP,IVIN,IVOUT,
659.      3ITCNT,INVF=0,ITPLIM,IFFEZ,JCOLP,NROW,NCOL,NLEEM,NETA,NLFLEM,NLETA,
660.      4NGELEM,NINF,NUELEM,NUEIA,NNEG0J,NLINES,ISTYPE(350),
661.      5LA(1302),LE(2002),PUN(8),
662.      6IPUNC,NDEGT,NDUAL,NIPIW,IFBAS,IFCRSH
663.      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
664.      COMMON IA(4000),IF(8000)
665.      COMMON/LP1/PI(1302),XX(1302)
666.      COMMON/LNCLNS/G1(350,10),G2(400,10),EA(350),BB(400)
667.      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
668.      COMMON/INDX2/ JH,DIGMA,KINGAS,IDBAS
669.      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
670.      EPS= 1E-13
671.      IF (KEY.EQ.1) GO TO 50
672.      IF (PD2.GT.0) GO TO 20
673.      KM= KMU
674.      KMU= KMU- 1
675.      IF (JS .GT. IH) GO TO 4
676.      MH= MUH(JS)
677.      MUH(JS)= 0
678.      IF (MH.EQ.KM) GO TO 6
679.      DO 5 I= MH,KMU
680.      IM= MU(I+1)
681.      MU(I)= IM
682.      MUH(IM)= MUH(IM)- 1
683.      DO 5 J= 1,N
684.      G2(J,I)= G2(J,I+1)
685.      5 CONTINUE
686.      MU(KM)= 0
687.      4 DO 7 J= 1,N
688.      G2(J,KM)= 0.0
689.      IF (KEY.EQ.2) RETURN
690.      GO TO 50
691.      20 KN= KNU
692.      KNU= KNU- 1
693.      IF (JS .GT. IH) GO TO 28
694.      NH= NUH(JS)
695.      NUH(JS)= 0
696.      IF (NH.EQ.KN) GO TO 26
697.      DO 25 I= NH,KNU
698.      JN= NU(I+1)
699.      NU(I)= JN
700.      NUH(JN)= NUH(JN)- 1
701.      DO 25 J= 1,M
702.      G1(J,I)= G1(J,I+1)
703.      25 CONTINUE
704.      NU(KN)= 0
705.      28 DO 30 I= 1,M
706.      G1(I,KN)= 0.0
707.      IF (KEY.EQ.2) RETURN

```

```

708. C      NOW WE ADD THE SUPERBASIC VARIABLE SPECIFIED BY PD1,IS.
709. C
710. 50 IF (PD1.LI.0) GO TO 70
711.    KN= KNU
712.    KNU= KNU+ 1
713.    IF (IS.GI.IH) GO TO 55
714.    NU(KNU)= IS
715.    NUH(IS)= KNU
716. 55 CALL UNPACK(IS)
717.    CALL FTRAN(1)
718.    DO 60 I= 1,M
719. 60 GI(I,KNU)= -Y(I)
720.    RETURN
721. C
722. C      NOW WE ARE BRINGING A DUAL SUPERBASIC COLUMN IN.
723. C
724. 70 KNU= KNU+ 1
725.    IF (IS.GI.IH) GO TO 73
726.    MU(KNU)= IS
727.    MUH(IS)= KNU
728. 73 IF (IS.LE.M) GO TO 110
729. C
730. C      THE ENTERING VARIABLE IS A ZETA.  THE ENTERING COLUMN
731. C      IS A ROW OF INV(B) AND A ROW OF INV(B)*D.
732. C
733.    IDD= KINBAS(IS)
734.    DO 75 I= 1,M
735. 75 Y(I)= 0.
736.    Y(IDD)= 1.
737.    CALL BTRAN
738.    CALL SHIFTR(3,2)
739.    DO 80 I= 1,M
740.    IF (IDEAS(I).EQ.0) GO TO 80
741.    GZ(IDBAS(I),KNU)= X(I)
742. 80 CONTINUE
743.    DO 100 J= MP1,NM
744.    IF (IDBAS(J).EQ.0) GO TO 100
745.    CALL UNPACK(J)
746.    DDT= 0.
747.    DO 90 I= 1,M
748.    IF (IDEAS(I).EQ.0) GO TO 90
749.    DDT= DDT+ Y(I)*X(I)
750. 90 CONTINUE
751.    GZ(IDBAS(J),KNU)= DDT
752. 100 CONTINUE
753.    RETURN
754. C
755. C      THE ENTERING VARIABLE IS A PI.  THE ENTERING COLUMN
756. C      IS PARTLY C1*INV(B) AND PARTLY C2+ C1*INV(B)*D.
757. C      -C1*INV(B) IS PART OF THE BASIS INVERSE CORRESPONDING TO
758. C      THE X-BASIC COLUMNS AND THE 1-BASIC ROWS.
759. C
760. 110 IDD= KINBAS(IS)
761.    DO 130 I= 1,M
762. 130 Y(I)= 0.
763.    Y(IDD)= 1.
764.    CALL BTRAN
765.    CALL SHIFTR(3,2)
766.    DO 140 I= 1,M
767.    IF (IDEAS(I).EQ.0) GO TO 140

```

```

768.      G2(IDBAS(I),KMU) = X(I)
769. 140 CONTINUE
770.      DO 160 J= MP1,NM
771.      IF (IDBAS(J).EQ.0) GO TO 160
772.      CALL UNPACK(J)
773.      DDT= 0.0
774.      DO 150 I= 1,M
775.      IF (IDBAS(I).EQ.0) GO TO 150
776.      DDT= DDT+ Y(I)*X(I)
777. 150 CONTINUE
778.      G2(IDBAS(J),KMU) = Y(15)+ DDT
779. 160 CONTINUE
780.      RETURN
781.      END
782.      SUBROUTINE RECALC
783.
784.      C
785.      C      THIS SUBROUTINE RECALCULATES THE SUPERBASIS COLUMNS
786.      C      USING THE CURRENT BASIS IN QR FORM. IT CAN ALSO ADD
787.      C      A VARIABLE AND COLUMN TO THE SUPERBASIS.
788.      C
789.      IMPLICIT REAL*8 (A-H,O-Z)
790.      INTEGER PD,PD1,PD2,JS,RP,ZFLAG,RS,P,DD,DD1,RL1,RL2
791.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
792.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(80)
793.      DOUBLE PRECISION E(2000)
794.      REAL A(4000),C(800),CMIN,COND,ERMAX,SUMINF
795.
796.      C
797.      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
798.      1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
799.      2NTEMP(20),KINP,ITIM,ITINV,ITINV,MSTAT,IOBJ,IRGWP,IVIN,IVOUT,
800.      3ITCNT,INVERQ,ITRLIM,IFEEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
801.      4NGELEM,NINF,NULEM,NUETA,NNEGDU,NLINES,ISTYPE(350),
802.      5LA(1302),LE(2002),PUN(8),
803.      6IPUNC,NDECI,NDUAL,NIPW,IFBAS,IFCRSH
804.      COMMON ITC,ITCA,IFPIW1,IFNEG,KOUTR
805.      COMMON IA(4000),IE(8000)
806.      COMMON/LP1/PI(1302),XX(1302)
807.      COMMON/UNCENS/G1(350,10),G2(400,10),EA(350),PB(400)
808.      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
809.      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
810.      COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
811.      COMMON/DIM/ IH,N,M,DD,KMU,KNU,RL1,RL2,MP1,NM,INO,IND
812.      EPS= 16.**(-13)
813.      IF (KNU.EQ.0) GO TO 26
814.      DO 28 J= 1,KNU
815.      IDO= NU(J)
816.      CALL UNPACK(IDO)
817.      CALL FTRAN(1)
818.      DO 25 I= 1,M
819.      G1(I,J)= - Y(I)
820.      25 CONTINUE
821.      CALL SHIFTR(1,3)
822.      CALL FTRAN(1)
823.      DO 22 I= 1,M
824.      BA(I)= Y(I)
825.      22 IF (KMU.EQ.0) GO TO 140
826.      DO 130 J= 1,KMU
827.      IDO= MU(J)
828.      IF (IDO.LE.M) GO TO 70

```



```

828. C      ZETA- VARIABLE IS ENTERING
829. C
830.      IB= KINBAS(100)
831.      DO 35 I= 1,M
832. 35 Y(I)= 0.
833.      Y(IB)= 1.
834.      CALL BTRAN
835.      CALL SHIFTR(3,2)
836.      DO 40 I= 1,M
837.      IF (IDBAS(I).EQ.0) GO TO 40
838.      G2(IDBAS(I),J)= X(I)
839. 40 CONTINUE
840.      DO 60 K= MP1,NM
841.      IF (IDBAS(K).EQ.0) GO TO 60
842.      CALL UNPACK(K)
843.      DOT= 0.0
844.      DO 50 I= 1,M
845.      IF (IDBAS(I).EQ.0) GO TO 50
846.      DOT= DOT+ Y(I) * X(I)
847. 50 CONTINUE
848.      G2(IDBAS(K),J)= DOT
849. 60 CONTINUE
850.      GO TO 150
851. C
852. C      PI- VARIABLE IS ENTERING
853. C
854. 70 IB= KINBAS(100)
855.      DO 90 I= 1,M
856. 90 Y(I)= 0.
857.      Y(IB)= 1.
858.      CALL BTRAN
859.      CALL SHIFTR(3,2)
860.      DO 100 I= 1,M
861.      IF (IDBAS(I).EQ.0) GO TO 100
862.      G2(IDBAS(I),J)= X(I)
863. 100 CONTINUE
864.      DO 120 K= MP1,NM
865.      IF (IDBAS(K).EQ.0) GO TO 120
866.      CALL UNPACK(K)
867.      DOT= 0.0
868.      DO 110 I= 1,M
869.      IF (IDBAS(I).EQ.0) GO TO 110
870.      DOT= DOT+ Y(I) * X(I)
871. 110 CONTINUE
872.      G2(IDBAS(K),J)= Y(100)+ DOT
873. 120 CONTINUE
874. 130 CONTINUE
875.      DO 150 I= 1,M
876. 150 Y(I)= 0.0
877.      IF (KINBAS(NM).NE.0) Y(KINBAS(NM))= 1.
878.      CALL BTRAN
879.      CALL SHIFTR(3,2)
880.      DO 160 I= 1,M
881. 160 IF (IDBAS(I).NE.0) B8(IDBAS(I))= X(I)
882.      DO 170 J= MP1,NM
883.      IF (IDBAS(J).EQ.0) GO TO 170
884.      CALL UNPACK(J)
885.      DOT= 0.0
886.      DO 165 I= 1,M
887.      IF (IDBAS(I).EQ.0) GO TO 165

```

```

888.      DOT= DOT+ X(I)*Y(I)
889. 165 CONTINUE
890.      BB(ICBAS(J))= DOT
891. 170 CONTINUE
892.      RETURN
893.      END
894.      SUBROUTINE ENDPNT(JS,PD1,IS,NET)
895.      IMPLICIT REAL*8 (A-H,O-Z)
896.      REAL*8 MIN,MIN2
897.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(H0)
898.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
899.      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,P,DD,DD1,BL1,BL2
900.      REAL C(400)
901.      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10)
902.      COMMON/LP1/PI(1302),XX(1302)
903.      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
904.      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
905.      COMMON/LNCNS/G1(350,10),G2(400,10),EA(350),HB(400)
906.      COMMON/INDX1/ NUH(10),NUH(10),NU(10)
907.      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
908.      COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
909.      COMMON/TIM/ IH,N,M,DD,KMU,KMU,BL1,BL2,MP1,NM,INQ,INP
910.      COMMON/INT/ IPS(30),KDET,KOUNT,ISING
911.      COMMON/TOLFR/ TOLF7,TOLCD,TOLCV,STPMX,STPRD
912.      DIMENSION U(3),V1(10),V2(10),F(10),DCT(4),RHS(10),UL(10,10)
913.
914.      C      THIS PROGRAM WILL FIND THE ENDPOINT OF A CURVE
915.      C      DEFINED BY A SYSTEM OF DD-1 BILINEAR EQUATIONS IN DD
916.      C      UNKNOWN. ONE ENDPOINT IS KNOWN, AND THE OTHER END-
917.      C      POINT IS DETERMINED BY THE FIRST INTERSECTION OF THE
918.      C      CURVE WITH ONE OF NM+1 INEQUALITY CONSTRAINTS.
919.      C      THE SYSTEM IS
920.      C
921.      C      BF1+ D2*X(MU)- DIAG(X(MU))*(F1*X(NU)+ E1)= 0
922.      C      BF2+ D2*X(MU)- DIAG(X(MU))*(F2*X(MU)+ E2)= 0
923.      C
924.      C      WHERE ONE OF THE ABOVE IS AN INEQUALITY IF ZFLAG.NE.1/
925.      C      THE INEQUALITY IS THE LAST IN THE GROUP DEFINED BY INEQ.
926.      C      THE LINEAR INEQUALITIES ARE DEFINED BELOW.
927.      C
928.      NET= 0
929.      FRAC=1.
930.      C      MFLAG=0 WHEN WE ARE SOLVING FOR THE INITIAL 3 POINTS
931.      C      MFLAG=1 WHEN ONE OF THE G-CONSTRAINTS IS BINDING.
932.      C      MFLAG=2 WHEN ONE OF THE VARIABLES GOES TO ZERO.
933.      C      MFLAG=3 WHEN WE USE THE NORMAL HYPERPLANE TO DEFINE THE
934.      C      SUBPROBLEM SHORT OF THE BOUNDARY OF THE CELL.
935.      ITER= 0
936.      ISGCT= 0
937.      MFLAG=0
938.      L=1
939.      KK= 0
940.      DD1= DD- 1
941.      ID= INQ
942.      KMU1= KMU+ 1
943.      IF (INEQ.EQ. 2) ID= INP+ KMU
944.      IF (PD.FG.0) ID= DD
945.      ICX= ID
946.      IF (KMU.FG.0) GO TO 615
947.      DO 610 I= 1,KMU

```

```

943.      610 X(I)= F1(MU(I))
944.      615 IF (KNU.EG.0) GO TO 625
945.      DO 620 I= 1,KNU
946.      KM= KML+ 1
947.      620 X(KM)= XX(MU(I))
948.      625 ALPHA= -1.
949.      5 ITER= ITER+ 1
950.      IF (DD.GT.1) GO TO 6
951.      KDET= 1
952.      GO TO 41
953.      6 IF (KNU.EG.DD .OR. KML.EG.DD) GO TO 630
954.      CALL DERIV
955.      7 ISGCT= ISGCT+ 1
956.      IF (ISGCT.GT.DD) GO TO 300
957.      DO 10 I= 1,DD1
958.      RHS(I)= -H(I,ICX)
959.      10 CONTINUE
960.      IF (ICX.EG.DD) GO TO 25
961.      DO 24 J= ICX,DD1
962.      JF1= J+ 1
963.      DO 23 I= 1,DD1
964.      F(I,J)= H(I,JF1)
965.      23 CONTINUE
966.      25 IF (DD1.GT.1) GO TO 35
967.      IF (DABS(H(1,1)) .GE. TOLCV) GO TO 26
968.      ISING= 1
969.      GO TO 37
970.      26 V2(1)= RHS(1)/H(1,1)
971.      IF (H(1,1).CE.0) GO TO 33
972.      KDET= -1
973.      GO TO 40
974.      33 KDET= 1
975.      GO TO 40
976.      35 CALL DCCOMP(DD1,F,UL)
977.      37 IF (ISING.NE.1) GO TO 38
978.      IF (ICX.EG. 1) GO TO 36
979.      ICX= ICX- 1
980.      GO TO 7
981.      36 ICX= DD
982.      GO TO 7
983.      38 CALL SOLVE(DD1,UL,RHS,V2)
984.      40 DO 39 I= 1,DD1
985.      IM= DD- I
986.      IF (IM.LT.ICX) GO TO 41
987.      IF1= IM+ 1
988.      V2(IF1)= V2(IM)
989.      39 CONTINUE
990.      GO TO 41
991.      630 KDET= 1
992.      DO 634 I= 1,DD
993.      V2(I)= 0.0
994.      634 CONTINUE
995.      41 V2(ICX)= 1.0
996.      IF (ITER.GT.1) GO TO 45
997.      SUM= 0.0
998.      IF (PD.EG.0) GO TO 120
999.      IF (PD.EG.-1) GO TO 110
1000.      JB= KINBAS(JS)
1001.      IF (JB.NE.0) GO TO 104
1002.      IF (V2(1D).GT.0) GO TO 135

```



```

1008.          GO TO 140
1009. 104 DO 105 I= 1,KMU
1010.      IK= KMU+ 1
1011.      SUM= SUM+ V2(IK)*G1(JB,I)
1012. 105 CONTINUE
1013.      IF (SUM.LT.0.0) GO TO 140
1014.      GO TO 135
1015. 110 JB= ICEAS(JS)
1016.      IF (JB.NE.0) GO TO 112
1017.          IF (V2(ID).GT.0) GO TO 135
1018.          GO TO 140
1019. 112 DO 115 I= 1,KMU
1020.      SUM= SUM+ V2(I)*G2(JB,I)
1021. 115 CONTINUE
1022.      IF (SUM.LT. 0.0) GO TO 140
1023.      GO TO 135
1024. C
1025. C      CALCULATE THE GRADIENT OF THE SURFACE, F(DD)= 0.0.
1026. C
1027. 120 IF (INQ.EQ.2) GO TO 126
1028. C      F(DD)= BF1(INQ)+ D1(INQ,.)*PI(MU)- PI(INQ)*
1029. C      (F1(INQ,.)**XX(NL)+ F1(INQ)).
1030.      INQ= MUH(DD)
1031.      DO 122 I= 1,KMU
1032. 122      V1(I)= D1(INQ,I)
1033.      DO 123 J= 1,KMU
1034.          JK= J+ KMU
1035.          SUM= SUM+ F1(INQ,J)*X(JK)
1036. 123      V1(INQ)= V1(INQ)- SUM- E1(INQ)
1037.      PID= X(INQ)
1038.      DO 124 J= 1,KMU
1039.          JK= J+ KMU
1040. 124      V1(JK)= - PID*F1(INQ,J)
1041.      SUM= 0.0
1042.      DO 125 I= 1,DD
1043. 125      SUM= SUM+ V1(I)*V2(I)
1044.      IF (SUM.LT.0) GO TO 140
1045.      GO TO 135
1046. C
1047. C      F(DD)= BF2(INQ)+ D2(KMU,.)*PI(MU)- X(IPQ)*(F2(INP,.)*PI(MU)+E2)
1048. C
1049. 126 IB= INP+ KMU
1050.      DO 127 J= 1,KMU
1051. 127      SUM= SUM+ F2(INP,J)*X(J)
1052.      PID= X(IB)
1053.      DO 128 J= 1,KMU
1054.          V1(J)= D2(INP,J)- PID*F2(INP,J)
1055. 128 CONTINUE
1056.      DO 129 J= KMU1,DD
1057. 129      V1(J)= 0.0
1058.      V1(IB)= -SUM-E2(INP)
1059.      SUM= 0.0
1060.      DO 130 I= 1,DD
1061. 130      SUM= SUM+ V1(I)*V2(I)
1062.      IF (SUM.LT.0) GO TO 140
1063. 135 INIT= KCET
1064.      GO TO 45
1065. 140 INIT= -KCET
1066. 45 CALL NCRM(V2,S1,DD)
1067.      S1= (KCET*INIT)/S1

```

```

1068.      DO 47 I= 1,DD
1069.          ACC(2,I)= V2(I)*S1
1070.          ACC(3,I)= X(I)
1071.      47 CONTINUE
1072.      WRITE (6,850)
1073.      WRITE (6,855) ((ACC(1,J),I=2,3),J=1,DD)
1074.
1075.      C      FIND THE FIRST BOUNDARY THAT THE TANGENT, Q(ALPHA)
1076.      C      HITS. WE ARE TRYING TO FIND
1077.      C      THE SMALLEST ALPHA SUCH THAT
1078.      C      G(1,.) * Q(ALPHA) + GE(1) = 0
1079.      C
1080.      IFLAG= 0
1081.      IF (KNU.EQ.0) GO TO 249
1082.      DO 250 I= 1,M
1083.          IV= JH(I)
1084.          IF (IV.EQ.M .OR. IV.LE.IH) GO TO 250
1085.          DO 240 LL= 2,3
1086.              U(LL)= 0.0
1087.              DO 230 K= 1,KNU
1088.                  KJ= K+ KNU
1089.                  U(LL)= U(LL)+ G1(I,K)*ACC(LL,KJ)
1090.      230      CONTINUE
1091.      240      CONTINUE
1092.      241      U(3)= U(3)+ EA(1)
1093.              IF (DABS(U(2)).LT.TOLFZ) GO TO 250
1094.              ALPHA= -U(3)/U(2)
1095.              IF (ALPHA .LT. -TOLFZ) GO TO 250
1096.              IF (ALPHA .GT. TOLBD) GO TO 244
1097.              TCD= 0.0
1098.              DO 242 K= 1,KNU
1099.                  KM= K+ KNU
1100.                  TCD= TCD+ G1(I,K)*ACC(2,KM)
1101.                  IF (TCD .GT. -TOLFZ) GO TO 250
1102.      244      IF (IFLAG.EQ.1) GO TO 243
1103.              IFLAG= 1
1104.              JJ= I
1105.              MINE= ALPHA
1106.              MPD= 1
1107.              GO TO 250
1108.      243      IF (ALPHA.GE.MIN) GO TO 250
1109.              JJ= I
1110.              MINE= ALPHA
1111.              MPD= 1
1112.      250      CONTINUE
1113.      249      IF (KMU.EQ.0) GO TO 640
1114.      DO 259 I= 1,N
1115.          IF (DICMA(I).EQ. M) GO TO 259
1116.          DO 256 LL= 2,3
1117.              U(LL)= 0.0
1118.              DO 254 K= 1,KMU
1119.                  U(LL)= U(LL)+ G2(I,K)*ACC(LL,K)
1120.      254      CONTINUE
1121.      256      CONTINUE
1122.      257      U(3)= U(3)+ FB(1)
1123.              IF (DABS(U(2)).LT.TOLFZ) GO TO 259
1124.              ALPHA= -U(3)/U(2)
1125.              IF (ALPHA .LT. -TOLFZ) GO TO 259
1126.              IF (ALPHA .GT. TOLBD) GO TO 258
1127.              TCE= 0.0

```

```

1128.          DO 252 K= 1,KMU
1129.          TOD= TOD+ ACC(2,K)*G2(1,K)
1130.          252      CCNTINUE
1131.          IF (TOD .GT. -TOLCZ) GO TO 259
1132.          258      IF (IFLAG.FO.1) GO TO 253
1133.          IFLAG = 1
1134.          JJ= 1
1135.          MIN= ALPHA
1136.          MPD= -1
1137.          GO TO 259
1138.          253      IF (ALPHA.GE.MIN) GO TO 259
1139.          JJ= 1
1140.          MIN= ALPHA
1141.          MPD= -1
1142.          259      CCNTINUE
1143.          640      MFLAG= 1
1144.          IMAG= 0
1145.          DO 588 I= 1,4
1146.          588      DOT(1)= 0.0
1147.          590      IF (INQ.EC.2) GO TO 600
1148.          DO 592 I= 1,DD
1149.          IF (I.GT.KMU) GO TO 591
1150.          DOT(1)= DOT(1)+ C1(INQ,I)*ACC(3,I)
1151.          DOT(2)= DOT(2)+ D1(INQ,I)*ACC(2,I)
1152.          GO TO 592
1153.          591      IK= I- KMU
1154.          DOT(3)= DOT(3)+ F1(INQ,IK)*ACC(3,I)
1155.          DOT(4)= DOT(4)+ F1(INQ,IK)*ACC(2,I)
1156.          592      CCNTINUE
1157.          U(1)= -ACC(2,INQ)*DOT(4)
1158.          U(2)= -(ACC(2,INQ)*E1(INQ)+ ACC(3,INQ)*DOT(4))+ DOT(2)
1159.          U(3)= BE1(INQ)+ DOT(1)- ACC(3,INQ)*(DOT(3)+ F1(INQ))
1160.          CALL QUADS(U,IMAG,ALPHA,BETA)
1161.          IF (ALPHA.LT.TOLCV) ALPHA= BETA
1162.          GO TO 605
1163.          600      IB= INP+ KMU
1164.          IF (KMU.EC.2) GO TO 603
1165.          DO 602 I= 1,KMU
1166.          DOT(1)= DOT(1)+ C2(INP,I)*ACC(3,I)
1167.          DOT(2)= DOT(2)+ D2(INP,I)*ACC(2,I)
1168.          DOT(3)= DOT(3)+ F2(INP,I)*ACC(3,I)
1169.          DOT(4)= DOT(4)+ F2(INP,I)*ACC(2,I)
1170.          602      CCNTINUE
1171.          603      U(1)= -ACC(2,IB)*DOT(4)
1172.          U(2)= -(ACC(2,IB)*E2(INP)+ ACC(3,INP)*DOT(4))+ DOT(2)
1173.          U(3)= BE2(INP)+ DOT(1)- ACC(3,INP)*(DOT(3)+ F2(INP))
1174.          CALL QUADS(U,IMAG,ALPHA,BETA)
1175.          IF (ALPHA.LT.TOLCV) ALPHA= BETA
1176.          605      IF (IMAG.EC.1) GO TO 260
1177.          IF (BETA .LT. -TOLCV) GO TO 260
1178.          IF (ALPHA .LT. -TOLCV) ALPHA= BETA
1179.          IF (ALPHA .GT. MIN) GO TO 260
1180.          JJ= M+ 1
1181.          MIN= ALPHA
1182.          MPD= 0
1183.          C
1184.          C      NOW CHECK TO SEE IF ONE OF THE VARIABLES GOES TO ZERO
1185.          C      BEFORE ANY OF THE CONSTRAINTS BECOME TIGHT.
1186.          C
1187.          260      MFLAG= 1

```



```

1188.      DO 280 I=1,DD
1189.      DO 265 II=2,3
1190.      U(LL)= ACC(LL,I)
1191. 265 CONTINUE
1192.      IF (DABS(U(2)) .LT. TOLFZ) GO TO 280
1193.      ALPHA= -U(3)/U(2)
1194.      IF (ALPHA .GT. MIN .OR. ALPHA .LT. TOLFZ) GO TO 280
1195.      JJ = 1
1196.      MFLAG= 2
1197.      MIN= ALPHA
1198. 280 CONTINUE
1199.      WRITE (6,635) MIN, MPD, JJ
1200. 635 FORMAT(' STEPLENGTH= ',F15.6,' CONSTRAINT TYPE:(',I3,',',I3,',')')
1201.      IF (MIN .LT. STPMX) GO TO 255
1202.      MIN= MIN* STPRD
1203.      MFLAG= 3
1204.      WRITE (6,635) MIN, MPD, JJ
1205. 255 BT= 0.0
1206.      DO 290 I= 1,DD
1207.      X(I)= MIN * ACC(2,I) + ACC(3,I)
1208.      BT= BT + X(I)*ACC(2,I)
1209. 290 CONTINUE
1210. C
1211. C      NEWTON'S METHOD FOR THE PROBLEM OF FINDING THE INTERSECTION
1212. C      OF THE CURVE WITH THE CONSTRAINT DEFINED BY MPD AND JJ.
1213. C
1214. 295 KK= KK+ 1
1215.      IF (KK.GE.5) STOP
1216.      DO 100 K= 1,15
1217.      KM1= K- 1
1218.      IRWDIM= DD1
1219.      IF(MFLAG.GE.1) IRWDIM= DD
1220.      CALL FTN(F,X)
1221.      CALL NORM(F,S1,IRWDIM)
1222.      WRITE (6,54) KM1,(X(I),I=1,DD)
1223.      FORMAT (' ITERATION',I3,' X= ',9F10.5)
1224. 54 WRITE (6,936) S1, (F(I),I=1,IRWDIM)
1225.      IF (S1.LT.TOLCV) GO TO 800
1226.      CALL CERIV
1227.      NET= NET+ 1
1228.      IF (DD.GT.1) GO TO 55
1229.      IF (DABS(H(1,1)) .LT. TOLCV) GO TO 58
1230.      Z(1)= F(1)/H(1,1)
1231.      GO TO 65
1232. 55 CALL DECOMP(IRWDIM,F,UL)
1233.      IF (ISING.FG.0) GO TO 60
1234. 58 WRITE (6,940)
1235.      STOP
1236. 60 CALL SOLVE(IRWDIM,UL,F,Z)
1237. 65 ALPH= 1.0
1238.      CALL CSENT(ALPH,S1)
1239.      IF (ALPH.GT. TOLFZ) GO TO 70
1240.      WRITE (6,945)
1241.      GO TO 800
1242. 70 DO 95 I= 1,DD
1243.      X(I)= X(I)- ALPH * Z(I)
1244. 95 CONTINUE
1245. 100 CONTINUE
1246.      MFLAG= 4
1247.      MIN= MIN/2.

```

```

1248.      IF (MIN.GT.TOLBD) GO TO 255
1249.      GO TO 300
1250. 800 IF (MFLAG.EQ.3) GO TO 5
1251.      CALL CONCHK(GMIN,KGMIN,MP2)
1252.      IF (KMU.EQ.0) GO TO 146
1253.      DO 145 I= 1,KMU
1254.          KM= NU(I)
1255.          PI(KM)= X(I)
1256.          IF (X(I).GE.GMIN) GO TO 145
1257.          GMIN= X(I)
1258.          KGMIN= KM
1259.          MP2= -1
1260. 145 CONTINUE
1261. 146 IF (KMU.EQ.0) GO TO 149
1262.      DO 147 I= KMU1,DC
1263.          IK= I- KMU
1264.          KM= NU(IK)
1265.          XX(KM)= X(I)
1266.          IF (X(I).GE.GMIN) GO TO 147
1267.          GMIN= X(I)
1268.          KGMIN= KM
1269.          MP2= 1
1270. 147 CONTINUE
1271. 148 IF (GMIN .LT. -TCLBD) GO TO 170
1272.      IF (MPD.NE. 0) GO TO 152
1273.      PD1= 0
1274.      RETURN
1275. 153 IF (MFLAG.EQ.2) GO TO 150
1276.      IF (MPD.EQ. -1) GO TO 151
1277.      IS= JH(JJ)
1278.      PD1= 1
1279.      RETURN
1280. 151 IS= DICMA(JJ)
1281.      PD1= -1
1282.      RETURN
1283. 150 IF (JJ.GT.KMU) GO TO 152
1284.      IS= NU(JJ)
1285.      PD1= -1
1286.      RETURN
1287. 152 JJ= JJ- KMU
1288.      IS= NU(JJ)
1289.      PD1= 1
1290.      RETURN
1291. C
1292. C      IF THE CURRENT VALUE OF X VIOLATES THE CONSTRAINT DEFINED
1293. C      BY KGMIN AND MP2, FIND A GOOD STARTING POINT FOR NEWTON'S
1294. C      METHOD BY FINDING THE POINT ON THE LINE SEGMENT
1295. C      ACC(1,..) + ALPHA*(X - ACC(1,..)), 0<= ALPHA<= 1
1296. C      THAT SATISFIES THE VIOLATED CONSTRAINT EXACTLY.
1297. C
1298. 170 JJ= KGMIN
1299.      MPD= MP2
1300.      MFLAG= 1
1301.      WRITE (6,174) MPD,JJ,GMIN
1302. 174 FORMAT(/,' THE QUADRATIC PICKED THE WRONG CONSTRAINT.'/
1303. 1,' THE MOST INFEASIBLE CONSTRAINT IS TYPE ',I3,
1304. 2,' NUMBER ',I4,/, ' WITH A VALUE OF ',F14.6)
1305.      DO 171 I= 1,DC
1306. 171      VI(I)= X(I)- ACC(3,I)
1307.      DO 175 I= 1,4

```

```

1308.      175      DOT(1)= 0.0
1309.      IF (MPD.EQ.0) GO TO 190
1310.      IF (MPD.EQ.-1) GO TO 180
1311.      DO 172 I= 1,KMU
1312.          IK= I+ KMU
1313.          DOT(1)= DOT(1)+ C1(JJ,I)*ACC(3,IK)
1314.          DOT(2)= DOT(2)+ C1(JJ,I)*V1(IK)
1315.      172 CONTINUE
1316.      STR= (-BA(JJ)-DOT(1))/DOT(2)
1317.      DO 173 I= 1,DD
1318.      173      X(I)= ACC(3,I)+ STR*V1(I)
1319.      GO TO 295
1320.      180 DO 183 I= 1,KMU
1321.          DOT(1)= DOT(1)+ G2(JJ,I)*ACC(3,I)
1322.          DOT(2)= DOT(2)+ G2(JJ,I)*V1(I)
1323.      183 CONTINUE
1324.      STR= (-BB(JJ)-DOT(1))/DOT(2)
1325.      DO 186 I= 1,DD
1326.      186      X(I)= ACC(3,I)+ STR*V1(I)
1327.      GO TO 295
1328.      C
1329.      C      THE BILINEAR CONSTRAINT IS NOT SATISFIED AT X. FIND A GOOD
1330.      C      STARTING POINT BY SOLVING THE QUADRATIC DEFINED BY THE INTER-
1331.      C      SECTION OF THE SEGMENT DEFINED ABOVE AND THE CONSTRAINT.
1332.      C
1333.      190 IF (INEQ.EQ.2) GO TO 200
1334.      DO 192 I= 1,DD
1335.          IF (I.GT.KMU) GO TO 191
1336.          DOT(1)= DOT(1)+ C1(ING,I)*ACC(3,I)
1337.          DOT(2)= DOT(2)+ C1(ING,I)*V1(I)
1338.          GO TO 192
1339.      191      IK= I- KMU
1340.          DOT(3)= DOT(3)+ F1(ING,IK)*ACC(3,I)
1341.          DOT(4)= DOT(4)+ F1(ING,IK)*V1(I)
1342.      192 CONTINUE
1343.      U(1)= -V1(ING)*DOT(4)
1344.      U(2)= -(V1(ING)*E1(ING)+ ACC(3,ING)*DOT(4))+ DOT(2)
1345.      U(3)= BF1(ING)+ DOT(1)- ACC(3,ING)*(DOT(3)+ E1(ING))
1346.      CALL QUADS(U,IMAG,STR,BETA)
1347.      IF (STR.LT.TOLCV) STR= BETA
1348.      DO 193 I= 1,DD
1349.      193      X(I)= ACC(3,I)+ STR*V1(I)
1350.      GO TO 295
1351.      200 IB= INP+ KMU
1352.      IF (KMU.EQ.0) GO TO 204
1353.      DO 202 I= 1,KMU
1354.          DOT(1)= DOT(1)+ B2(INP,I)*ACC(3,I)
1355.          DOT(2)= DOT(2)+ B2(INP,I)*V1(I)
1356.          DOT(3)= DOT(3)+ F2(INP,I)*ACC(3,I)
1357.          DOT(4)= DOT(4)+ F2(INP,I)*V1(I)
1358.      202 CONTINUE
1359.      204 U(1)= -V1(IB)*DOT(4)
1360.      U(2)= -(V1(IB)*E2(INP)+ ACC(3,INP)*DOT(4))+ DOT(2)
1361.      U(3)= BF2(INP)+ DOT(1)- ACC(3,INP)*(DOT(3)+ E2(INP))
1362.      CALL QUADS(U,IMAG,STR,BETA)
1363.      IF (STR.LT.TOLCV) STR= BETA
1364.      DO 203 I= 1,DD
1365.      203      X(I)= ACC(3,I)+ STR*V1(I)
1366.      GO TO 295
1367.      300 WRITE (6,980)

```



```

1368.      850 FORMAT(1X,'THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:'
1369.      1,/,1X,'          A2(*U)+          A3')
1370.      855 FORMAT (1X,2F17.7)
1371.      935 FORMAT(1X,'NORM(F(X))= ',F10.6,' F= ',9F10.6)
1372.      940 FORMAT (1X,/'*****ALGORITHM BONDED WITH SINGULAR JACOBIAN*****')
1373.      945 FORMAT(1X,'NO DESCENT POSSIBLE -- ALPHA WAS ZERO.')
```

1374. 980 FORMAT(1X,'NEWTON'S METHOD FAILED TO CONVERGE.')

1375. STOP

1376. END

1377. C QUADS FINDS THE SMALLEST NONNEGATIVE ROOT OF

1378. C $U(1)*ALPHA**2 + U(2)*ALPHA + U(3) = 0$, IF THERE

1379. C IS ONE. IF NOT, IMAG IS SET = 1.

1380. C

1381. SUBROUTINE QUADS(U,IMAG,ALPHA,BETA)

1382. IMPLICIT REAL*8 (A-H,O-Z)

1383. INTEGER PD,PD1,PD2,SS,FR,ZFLAG,P,DD,DD1,PL1,PL2

1384. INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)

1385. COMMON/TOLER/ TOLFZ,TOLRD,TOLCV,STPMX,STPRD

1386. COMMON/DIM/ IH,N,M,DD,KMU,KNU,PL1,PL2,MP1,NM,INQ,INP

1387. DIMENSION U(3)

1388. IMAG= 0

1389. IF (DABS(U(1)).GT. TOLFZ) GO TO 1)

1390. IF (DABS(U(2)).LT.TOLFZ) GO TO 8

1391. ALPHA = -U(3)/U(2)

1392. BETA= ALPHA

1393. RETURN

1394. 8 IMAG= 1

1395. RETURN

1396. 1) RT= U(2)**2 - 4*U(1)*U(3)

1397. IF (RT) 20, 30, 40

1398. C

1399. C THE QUADRATIC DOES NOT INTERSECT THIS CONSTRAINT.

1400. C

1401. 20 IMAG= 1

1402. RETURN

1403. C

1404. C THERE IS A DOUBLE ROOT.

1405. C

1406. 30 ALPHA = U(2)/(U(1)*2.)

1407. BETA= ALPHA

1408. RETURN

1409. C

1410. C TWO REAL ROOTS EXIST.

1411. C

1412. 40 DIS= DSQRT(RT)

1413. ALPHA= (-U(2)-DIS)/(2.*U(1))

1414. BETA= (-U(2)+DIS)/(2.*U(1))

1415. IF (ALPHA .LE. BETA) RETURN

1416. SAVE= ALPHA

1417. ALPHA = BETA

1418. BETA= SAVE

1419. RETURN

1420. END

1421. SUBROUTINE CONCHK(GMIN,KGMIN,MP2)

1422. C

1423. C THIS SUBROUTINE EVALUATES THE CONSTRAINT FUNCTIONS

1424. C AND FINDS THE SMALLEST VALUE IN GMIN. ZFLAG IS BOTH AN

1425. C INPUT AND OUTPUT PARAMETER. IF ZFLAG.EC.1 INITIALLY, THEN

1426. C ONLY THE NONLINEAR CONSTRAINT IS EVALUATED. ZFLAG IS SET

1427. C EQUAL TO 2 IF ANY CONSTRAINT IS NONPOSITIVE, OTHERWISE IT

```

1428. C REMAINS ZERO.
1429. C
1430. IMPLICIT REAL*8 (A-H,O-Z)
1431. INTEGER PD,PD1,PD2,SS,RI,ZFLAG,P,DD,DD1,RL1,RL2
1432. INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
1433. REAL A(4000),C(800),CMIN,COND,ERMAX,SUMINF
1434. INTEGER*2 JH(350),DIGMA(552),KINEAS(1302),IDBAS(1312)
1435. COMMON/NFAT/ H(10,11),X(10),Z(10),ACC(3,10)
1436. COMMON/BLCST/BF1(10),BF2(10),F1(10),F2(10),C,IC,LC
1437. COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
1438. COMMON/CLP1/P1(1302),XX(1302)
1439. COMMON/LNCENS/G1(350,10),G2(400,10),RA(350),RB(400)
1440. COMMON/INDX2/JH,DIGMA,KINEAS,IDBAS
1441. COMMON/SCAL/ BT,NR,JJ,MELAG,DD1,P,PD,MPD,INEO,KFUN,KJAC
1442. COMMON/DIM/ IH,N,M,DD,KMU,KNU,RL1,RL2,MD1,NM,INO,INP
1443. GMIN= 10.
1444. ZFLAG= 0
1445. DO 20 I= 1,M
1446. IK= JH(I)
1447. GG= 0.0
1448. IF (KNU.EQ.0) GO TO 15
1449. DO 10 J= 1,KNU
1450. JK= J+ KMU
1451. GG= GG+ G1(I,J)*X(JK)
1452. 10 CONTINUE
1453. 15 GG= GG+ RA(I)
1454. XX(IK)= GG
1455. IF (IK.EQ. N) GO TO 20
1456. IF (GG.GE.GMIN) GO TO 20
1457. GMIN= GG
1458. KGMIN= I
1459. MP2= 1
1460. 20 CONTINUE
1461. DO 40 I= 1,N
1462. IK= DIGMA(I)
1463. GG= 0.0
1464. IF (KMU.EQ.0) GO TO 35
1465. DO 30 J= 1,KMU
1466. GG= GG+ G2(I,J)*X(J)
1467. 30 CONTINUE
1468. 35 GG= GG+ FB(I)
1469. FI(IK)= GG
1470. IF (GG.GE.GMIN) GO TO 40
1471. GMIN= GG
1472. KGMIN= I
1473. MP2= -1
1474. 40 CONTINUE
1475. IF (PD.EQ.0) RETURN
1476. 45 DOT= 0.
1477. DOT1= 0.
1478. GG= 0.0
1479. IF (INEO - 2) 50, 60, 60
1480. 50 IF (KNU.EQ.0) GO TO 55
1481. DO 54 J= 1,KNU
1482. JK= J+ KMU
1483. 54 DOT= DOT+ F1(INO,J)*X(JK)
1484. 55 GG= BF1(INO)- X(INO)*(DOT+ E1(INO))
1485. IF (KMU.EQ.0) GO TO 60
1486. DO 57 J= 1,KMU
1487. 57 GG= GG+ C1(INO,J)*X(J)

```

```

1488.      GO TO 87
1489.      60 KB2= KMU+ INP
1490.      IF (KMU.EQ.0) GO TO 65
1491.      DO 63 J= 1,KMU
1492.      DOT= DOT+ F2(INP,J)*X(J)
1493.      63 DOT1= DOT1+ D2(INP,J)*X(J)
1494.      65 GG= BF2(INP)+ DOT1- X(KP2)*(DOT+ F2(INP))
1495.      GO TO 86
1496.      80 IF (GG.GE.GMIN) RETURN
1497.      GMIN= GG
1498.      MP2= 0
1499.      RETURN
1500.      END
1501.      C
1502.      C      SUBROUTINE FTN EVALUATES THE FUNCTION F(X)=
1503.      C      <B(MU)+ B1*X(MU) + DIAG(X(MU))*(F1*X(MU)) >
1504.      C      <B(NU)+ D2*X(MU) + DIAG(X(NU))*(F2*X(MU)+ F3) >
1505.      C
1506.      C      SUBROUTINE FTN(F,Y)
1507.      C
1508.      C      THIS ROUTINE EVALUATES THE FUNCTION WHICH THE ENDPOINT
1509.      C      SUBROUTINE IS CURRENTLY TRYING TO FIND A ROOT OF. IF MFLAG
1510.      C      EQUALS 1 THEN THE LAST FUNCTIONAL COMPRISING F IS ONE OF THE
1511.      C      INEQUALITY CONSTRAINTS-- THE INEQUALITY DETERMINED BY THE
1512.      C      PARAMETERS MPD AND JJ.
1513.      C
1514.      IMPLICIT REAL*8 (A-H,C-Z)
1515.      INTEGER PD,PD1,PD2,SS,RP,ZFLAG,P,DD,DD1,BL1,BL2
1516.      INTEGER*2 ISTYPE,IA,LE,IA,IE,PUN,LC(20),IC(800)
1517.      REAL A(4000),C(800),CMIN,CEND,FRMAX,SUMINF
1518.      INTEGER*2 JH(35),DIGMA(952),KINEAS(1302),IDHAS(1302)
1519.      COMMON/NEW1/ H(10,11),X(10),Z(10),ACC(3,10)
1520.      COMMON/PLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
1521.      COMMON/PLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
1522.      COMMON/LNCCNS/G1(350,10),G2(400,10),FA(350),BB(400)
1523.      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
1524.      COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
1525.      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INO,INP
1526.      INTEGER BL1,BLJ,B2L,CDC
1527.      DIMENSION F(DD),Y(DD)
1528.      F(DD)= 0.
1529.      KFUN= KFUN+ DD
1530.      KMU1= KMU+ 1
1531.      IRD= 0
1532.      IF (KMU.EQ.0) GO TO 23
1533.      DD 20 I= 1,KMU
1534.      IF (INEQ.EQ.2) GO TO 5
1535.      IF (I.EQ.INO) GO TO 19
1536.      5 IR= I- 160
1537.      DOT= 0.
1538.      IF (KNU.EQ.0) GO TO 15
1539.      DO 10 J= 1,KNU
1540.      JK= J+ KMU
1541.      10 DOT= DOT+ F1(I,J)*Y(JK)
1542.      15 F(IR)= -Y(I)*(DOT+ E1(I))+ BF1(I)
1543.      IF (KMU.EQ.0) GO TO 20
1544.      DO 18 J= 1,KMU
1545.      18 F(IR)= F(IR)+ D1(I,J)*Y(J)
1546.      GO TO 20
1547.      19 IRD= 1

```

```

1548.      20 CONTINUE
1549.      23 IF (INL2.EQ.0) GO TO 70
1550.          IB= INP+ KMU
1551.          DO 40 J= KMU1,DD
1552.              IMB= I- KML
1553.              IF (INL2.EQ.1) GO TO 24
1554.              IF (I.EQ.1B) GO TO 35
1555.      24 IB= I- IBO
1556.          DOT= 0.0
1557.          IF (KMU.EQ.0) GO TO 30
1558.          DO 25 J= 1,KMU
1559.      25 DOT= DOT+ F2(IMB,J)*Y(J)
1560.      30 F(IP)= -Y(I)*(DOT+E2(IMB))+ BF2(IMB)
1561.          IF (KMU.EQ.0) GO TO 40
1562.          DO 32 J= 1,KMU
1563.      32 F(IR)= F(IP)+ D2(IMB,J)*Y(J)
1564.          GO TO 40
1565.      35 IBO= 1
1566.      40 CONTINUE
1567.      70 IF (MFLAG.LI.1) RETURN
1568.          IF (MPD.NE.0) GO TO 120
1569.          DOT= 0.
1570.          DOT1= 0.
1571.          DOT2= 0.
1572.          IF (INQ - 2) 80, 90, 90
1573.      80 IF (KMU.EQ.0) GO TO 85
1574.          DO 84 J= 1,KMU
1575.              JK= J+ KML
1576.      84 DOT= DOT+ F1(INQ,J)*Y(JK)
1577.      85 F(DD)= BF1(INQ)- Y(INQ)*(DOT+ E1(INQ))
1578.          IF (KMU.EQ.0) GO TO 120
1579.          DO 87 J= 1,KMU
1580.      87 F(DD)= F(DD)+ D1(INQ,J)*Y(J)
1581.          GO TO 120
1582.      90 IF (KMU.EQ.0) GO TO 95
1583.          DO 93 J= 1,KMU
1584.              DOT= DOT+ F2(INP,J)*Y(J)
1585.      93 DOT1= DOT1+ D2(INP,J)*Y(J)
1586.      95 IKN= INP+ KMU
1587.          F(DD)= BF2(INP)+ DOT1- Y(IKN)*(DOT+ E2(INP))
1588.      120 IF (MFLAG.LI.1) RETURN
1589.          IF (MPD.EQ.0) RETURN
1590.          IF (MFLAG - 2) 125,160,170
1591.      125 IF (MPD.EQ. -1) GO TO 140
1592.          IF (KMU.EQ.0) GO TO 135
1593.          DO 130 J= 1,KMU
1594.              JK= J+ KMU
1595.      130 F(DD)= F(DD)+ G1(JJ,J)*Y(JK)
1596.      135 F(DD)= F(DD)+ BA(JJ)
1597.          RETURN
1598.      140 IF (KMU.EQ.0) GO TO 150
1599.          DO 145 J= 1,KMU
1600.      145 F(DD)= F(DD)+ G2(JJ,J)*Y(J)
1601.      150 F(DD)= F(DD)+ EB(JJ)
1602.          RETURN
1603.      160 F(DD)= Y(JJ)
1604.          RETURN
1605.      170 DOT= 0.0
1606.          DO 180 J= 1,DD
1607.      180 DOT= DOT+ ACC(2,J)*Y(J)

```



```

1608.      F(DD)= DOT- RT
1609.      RETURN
1610.      END
1611.      SUBROUTINE DERIV
1612.      C
1613.      C THIS SUBROUTINE CALCULATES THE EXACT JACOBIAN OF THE
1614.      C BILINEAR FUNCTION DEFINED BY BLCONS. THERE MAY OR MAY
1615.      C NOT BE ANOTHER FUNCTIONAL APPENDED WHICH WE ARE ATTEMPT-
1616.      C ING TO MAKE BINDING WITH NEWTON'S METHOD. ALSO, A COL-
1617.      C UMN MAY BE OMITTED IF THE ENDPOINT ALGORITHM REQUIRES
1618.      C THE CORRESPONDING VARIABLE TO BE FIXED.
1619.      C
1620.      IMPLICIT REAL*8 (A-H,O-Z)
1621.      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,P,DD,DD1,BL1,BL2
1622.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
1623.      REAL A(4000),C(F00),CMIN,CCND,ERMAX,SUMINF
1624.      INTEGER*2 JF(350),DIGMA(952),KINEBAS(1302),IDRAS(1302)
1625.      COMMON/NEW1/ H(10,11),X(10),Z(10),ACC(3,10)
1626.      COMMON/HLCST/FF1(10),FF2(10),E1(10),E2(10),C,IC,LC
1627.      COMMON/HLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
1628.      COMMON/LNCONS/G1(350,10),G2(400,10),FA(350),BB(400)
1629.      COMMON/INX1/ NUH(10),MUH(10),NU(10),MU(10)
1630.      COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MDD,INEQ,KFUN,KJAC
1631.      COMMON/DIM/ IH,N,M,DD,KML,KNU,BL1,BL2,MP1,NM,INO,IND
1632.      INTEGER COLJJ
1633.      KJAC= KJAC+ DD * DD
1634.      IF (MFLAG.EQ.0) KJAC= KJAC- DD
1635.      DO 10 I= 1,DD
1636.          DO 10 J= 1,DD
1637.              10      H(I,J)= C.
1638.      KMU= KMU+1
1639.      KNU= KNU+ 1
1640.      DD1= DD-1
1641.      COLJJ= J
1642.      IF (KMU.EQ.0) GO TO 55
1643.      DO 50 J=1,KMU
1644.          IF (J.EQ.JJ .AND. MFLAG.GT.1) GO TO 45
1645.          JM= J-COLJJ
1646.          IF (KMU.EQ.0) GO TO 35
1647.          IRC= C
1648.          DO 30 I= 1,KMU
1649.              IF (INEQ.EQ.2) GO TO 15
1650.              IF (I.EQ.INQ) GO TO 28
1651.              15      IR= I- IRC
1652.              H(IR,JM)= D1(I,J)
1653.              IF (I.NE.J) GO TO 30
1654.              DOT= 0.0
1655.              IF (KNU.EQ.0) GO TO 25
1656.              DO 20 K= 1,KNU
1657.                  KB= KMU+K
1658.                  DOT= DOT+ F1(I,K)*X(KB)
1659.                  25      H(IR,JM)= H(IR,JM)- DOT- E1(I)
1660.                  GO TO 30
1661.                  28      IRC= I
1662.                  30      CONTINUE
1663.                  35      IF (F12.EQ.C) GO TO 50
1664.                  DO 40 I= KMU1,DD
1665.                      IMF= I- KMU
1666.                      IF (INEQ.EQ.1) GO TO 37
1667.                      IF (IMF.EQ.INP) GO TO 38

```

```

1668.      37      IR= I- IRC
1669.      H(IR,JM)= D2(IMB,J)- F2(IMB,J)*X(I)
1670.      GO TO 40
1671.      38      IRC= 1
1672.      40      CONTINUE
1673.      GO TO 50
1674.      45      COLJJ= 1
1675.      50      CONTINUE
1676.      55      IF (KMU.EC.0) GO TO 141
1677.      DO 70 J= KMU1,DD
1678.      IF (J.EQ.JJ .AND. MFLAG.GT.1) GO TO 65
1679.      JM= J-COLJJ
1680.      JMU= J-KMU
1681.      IF (KMU.EC.0) GO TO 72
1682.      IRC= 0
1683.      DO 60 I= 1,KMU
1684.      IF (INEQ.EC.2) GO TO 57
1685.      IF (I.EQ.1NQ) GO TO 58
1686.      57      IR= I- IRC
1687.      H(IR,JM)= - X(I)*F1(I,JMU)
1688.      GO TO 60
1689.      58      IRC= 1
1690.      60      CONTINUE
1691.      GO TO 70
1692.      65      COLJJ= 1
1693.      70      CONTINUE
1694.      COLJJ= 0
1695.      72      IRC= 0
1696.      IF (INEQ.EC.1) IRC= 1
1697.      IF (BL2.EC.0) GO TO 141
1698.      DO 80 I= 1,KMU
1699.      IKM= I+KMU
1700.      IF (IKM.GE.JJ .AND. MFLAG.GT.1) COLJJ= 1
1701.      IF (IKM.EC.JJ .AND. MFLAG.GT.1) GO TO 80
1702.      IF (INEQ.EC.1) GO TO 73
1703.      IF (I.EQ.1NQ) GO TO 78
1704.      73      IR= I- IRC
1705.      IK= KMU+IR
1706.      KNC= IKM - COLJJ
1707.      IF (KMU.EC.0) GO TO 77
1708.      DO 75 K= 1,KMU
1709.      75      H(IK,KNC)= H(IK,KNC)- F2(I,K)*X(K)
1710.      77      H(IK,KNC)= H(IK,KNC)- F2(I)
1711.      GO TO 80
1712.      78      IRC= 1
1713.      80      CONTINUE
1714.      C
1715.      C      IF MFLAG=1 THE INEQUALITY SPECIFIED BY MPD AND JJ
1716.      C      DETERMINES THE DDTH ROW OF THE JACOBIAN MATRIX H.
1717.      C
1718.      141      IF (MFLAG.LT.1) RETURN
1719.      IF (MPD.EC.0) GO TO 145
1720.      GO TO 130
1721.      145      IF (INEQ-2) 150,160,160
1722.      150      DO 155 J= 1,DD
1723.      IF (J.GT.KMU) GO TO 157
1724.      H(DD,J)= D1(1NQ,J)
1725.      IF (J.NE. 1NQ) GO TO 155
1726.      DD1= 0.0
1727.      IF (KMU.EC.0) GO TO 154

```

```

1728.      DO 152 K= 1,KNU
1729.      KB= KMU+ K
1730.      152 DOT= DOT+ F1(INQ,K)*X(KB)
1731.      154 H(DO,J)= F(DO,J)- DOT- E1(KMU)
1732.      GO TO 155
1733.      157 JK= J-KMU
1734.      H(DO,J)= -F1(INQ,JK)*X(INQ)
1735.      155 CONTINUE
1736.      GO TO 180
1737.      160 KB2= INP+ KMU
1738.      DO 164 J= 1,DO
1739.      IF (J.GT. KMU) GO TO 165
1740.      H(DO,J)= D2(INP,J)- F2(INP,J)*X(KB2)
1741.      GO TO 164
1742.      165 IF (J.NE. KB2) GO TO 164
1743.      DOT= 0.0
1744.      IF (KMU.EQ.0) GO TO 168
1745.      DO 167 K= 1,KMU
1746.      167 DOT= DOT+ F2(INP,K)*X(K)
1747.      168 H(DO,J)= -DOT-F2(INP)
1748.      164 CONTINUE
1749.      GO TO 180
1750.      180 IF (MFLAG.LT.1) RETURN
1751.      IF (MPD.EQ.0) RETURN
1752.      IF (MFLAG - 2) 190,230,240
1753.      190 IF (MPD.EQ. -1) GO TO 210
1754.      DO 200 J= KMU1,DO
1755.      JM= J- KMU
1756.      200 H(DO,J)= G1(JJ,JM)
1757.      RETURN
1758.      210 DO 220 J= 1,KMU
1759.      220 H(DO,J)= C2(JJ,J)
1760.      RETURN
1761.      230 H(DO,JJ)= 1.0
1762.      RETURN
1763.      240 DO 250 I= 1,DO
1764.      250 H(DO,I)= ACC(2,I)
1765.      RETURN
1766.      END
1767.      SUBROUTINE NORM(Y,S1,N)
1768.      IMPLICIT REAL*8 (A-H,C-Z)
1769.      DIMENSION Y(10)
1770.      S1= 0.0
1771.      DO 10 I= 1,N
1772.      10 S1= S1+ Y(I)*Y(I)
1773.      S1= DSQRT(S1)
1774.      RETURN
1775.      END
1776.      C
1777.      C      DSENT IS A SUBROUTINE WHICH IMPLEMENTS DESCENT ON
1778.      C      THE NORM OF THE FUNCTION WHOSE ROOT IS BEING DETER-
1779.      C      MINED BY NEWTON'S METHOD.
1780.      C
1781.      SUBROUTINE DSENT(ALPHA,ENCRM)
1782.      IMPLICIT REAL*8 (A-H,C-Z)
1783.      INTEGER PD,PD1,PD2,GS,RE,ZFLAG,P,DD,DD1,BL1,BL2
1784.      COMMON/DIM/ IH,N,M,DO,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
1785.      COMMON/NEW1/ H(10,11),X(10),Z(10),ACC(3,10)
1786.      DIMENSION Y(10),F(10)
1787.      1 DO 5 I=1,DO

```

```

1788.      5 Y(I)= X(I)- ALPHA*Z(I)
1789.      CALL FIN(F,Y)
1790.      CALL NORM(F,S2,DE)
1791.      IF (S2.LT.PNORM) RETURN
1792.      ALPHA= ALPHA/2.
1793.      IF (ALPHA.GT. .00001) GO TO 1
1794.      RETURN
1795.      END
1796.
1797.      C
1798.      C      DECOMP AND SOLVE ARE BORROWED FROM FORSYTHE'S THE SOLUTION
1799.      C      OF LINEAR SYSTEMS OF EQUATIONS. THEY IMPLEMENT A GAUSSIAN ELIM-
1800.      C      INATION SCHEME WITH PARTIAL PIVOTING TO PRODUCE A LU DECOMPOSITION
1801.      C      OF THE MATRIX A. 'SOLVE' PERFORMS THE BACK SUBSTITUTIONS NECESSARY
1802.      C      TO SOLVE      UL * X = B.      KDET FINDS SIGN( DET A ).
1803.      C
1804.      SUBROUTINE DECOMP(NN,A,UL)
1805.      IMPLICIT REAL*8(A-H,C-Z)
1806.      COMMON/INT/ IPS(30),KDET,KCNT,ISING
1807.      DIMENSION A(10,10),UL(10,10),SCALES(10)
1808.      N= NN
1809.      KDET= 1
1810.
1811.      C
1812.      C      INITIALIZE IPS,UL, AND SCALES.
1813.      ISING= 0
1814.      DO 5 I= 1,N
1815.      IPS(I)= I
1816.      RCWNRN= 0.0
1817.      DO 2 J= 1,N
1818.      UL(I,J)= A(I,J)
1819.      IF (RCWNRN-DAES(UL(I,J))) 1,2,2
1820.      RCWNRN= DABS(UL(I,J))
1821.      1 CONTINUE
1822.      IF (RCWNRN) 3,4,3
1823.      SCALES(I)= 1./RCWNRN
1824.      GO TO 5
1825.      4 CALL SING(1)
1826.      ISING= 1
1827.      SCALES(I)= 0.
1828.      5 CONTINUE
1829.
1830.      C
1831.      C      GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
1832.      NM1= N- 1
1833.      DO 17 K= 1,NM1
1834.      BIG= 0.0
1835.      DO 11 I= K,N
1836.      IF= IPS(I)
1837.      SIZE= DABS(UL(IP,K))*SCALES(IP)
1838.      IF (SIZE-BIG) 11,11,10
1839.      10 BIG= SIZE
1840.      ICXPIV= I
1841.      11 CONTINUE
1842.      IF (BIG) 13,12,13
1843.      CALL SING(2)
1844.      ISING= 1
1845.      GO TO 17
1846.      13 IF (ICXPIV-K) 14,15,14
1847.      J= IPS(K)
1848.      IPS(K)= IPS(ICXPIV)
1849.      IPS(ICXPIV)= J
1850.      KDET= -KDET

```



```

1348.      15      KP= IPS(K)
1349.             PIVOT= UL(KP,K)
1350.             KP1= K+1
1351.             DO 16 I= KP1,N
1352.                 IF= IPS(I)
1353.                 FM= -UL(IP,K)/PIVOT
1354.                 UL(IP,K)= -FM
1355.                 DO 16 J= KP1,N
1356.                     UL(IP,J)= UL(IP,J)+ FM*UL(KP,J)
1357.             16      CONTINUE
1358.             17      CONTINUE
1359.             KP= IPS(N)
1360.             IF (UL(KP,N)) 19,19,19
1361.             14      CALL SING(2)
1362.             ISING= 1
1363.             19      RETURN
1364.             END
1365.
1366. C
1367. C      SUBROUTINE SOLVE(NN,UL,B,X)
1368. C      IMPLICIT REAL*8(A-H,O-Z)
1369. C      COMMON/INT/ IPS(30),KDET,KCNT,ISING
1370. C      DIMENSION UL(10,10),B(10),X(10)
1371. C      N= NN
1372. C      NP1= N+ 1
1373. C      IP= IPS(1)
1374. C      X(1)= B(IP)
1375. C      DO 2 I= 2,N
1376. C          IP= IPS(I)
1377. C          IM1= I-1
1378. C          SUM= 0.0
1379. C          DO 1 J= 1,IM1
1380. C              SUM= SUM+ UL(IP,J)*X(J)
1381. C          2      X(I)= B(IP)- SUM
1382. C
1383. C          IP= IPS(N)
1384. C          X(N)= X(N)/UL(IP,N)
1385. C          IF(UL(IP,N).GE.0) GO TO 10
1386. C          KDET= -KDET
1387. C      10      DO 4 IBACK= 2,N
1388. C          I= NP1- IBACK
1389. C          1      GOES (N-1),.....1
1390. C              IP= IPS(I)
1391. C              IP1= I+1
1392. C              SUM= 0.0
1393. C              DO 3 J= IP1,N
1394. C                  SUM= SUM+ UL(IP,J)*X(J)
1395. C              3      DIV= UL(IP,I)
1396. C                  IF(DIV.GE.0) GO TO 4
1397. C                  KDET= -KDET
1398. C              4      X(I)= (X(I)-SUM)/DIV
1399. C              RETURN
1400. C          END
1401. C
1402. C      SUBROUTINE SING(IWHY)
1403. C      11      FORMAT(1X,'MATRIX WITH ZERO ROW IN DECOMPOSE.')
1404. C      12      FORMAT(1X,'SINGULAR MATRIX IN DECOMPOSE. ZERO DIVIDE IN SOLVE')
1405. C      IF (IWHY= 1) 1,1,2
1406. C      1      WRITE (6,11)
1407. C

```

```

1408.      GO TO 10
1409.      2 WRITE (6,121)
1410.      10 RETURN
1411.      END
1412.      SUBROUTINE DEBUG(MODE)
1413.      IMPLICIT REAL*8 (A-H,O-Z)
1414.      REAL*8 MIN
1415.      REAL*4 TIME
1416.      INTEGER PD,PD1,PD2,SS,RP,ZFLAG,RS,P,CD,DD1,HL1,RL2
1417.      INTEGER*2 JH(350),DIGMA(952),KINFAS(1302),IDPAS(1302)
1418.      INTEGER*2 ISTYLE,LA,LE,IA,IE,PUN,LC(20),IC(800)
1419.      DOUBLE PRECISION T(8000)
1420.      REAL A(4000),C(800),CMIN,COND,ERMAX,SUMINF
1421.
1422.      C
1423.      COMMON /SUM,OPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
1424.      1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
1425.      2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,I0BJ,IROWP,IVIN,IVOUT,
1426.      3ITCNT,INVERO,ITRLIM,IFFF7,JCCLP,NRCW,NCOL,NLEEM,NETA,NLELEM,NLFTA,
1427.      4NGELSM,NINF,NUELFN,NUETA,NNFGDJ,NLINES,ISTYPE(350),
1428.      5LA(1302),LE(2002),PUN(8),
1429.      6IPUNC,NSECI,NDUAL,NIP1W,IFEAS,IFCRSH
1430.      COMMON /ICG,ICHA,IFPIWT,IFNEG,KOUTB
1431.      COMMON /A(4000),IE(8000)
1432.      COMMON /LP1/PI(1302),XX(1302)
1433.      COMMON /BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
1434.      COMMON /RLC12/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
1435.      COMMON /LNCNS/G1(350,10),G2(400,10),FA(350),EB(400)
1436.      COMMON /INCX1/ NUH(10),MUH(10),NU(10),MU(10)
1437.      COMMON /INCX2/ JH,DIGMA,KINBAS,IDPAS
1438.      COMMON /SCAL/ RT,RP,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
1439.      COMMON /DIM/ IH,N,M,DD,KMU,KNU,RL1,RL2,MP1,NM,INQ,INP
1440.      IF (MODE-2) 10,30,60
1441.      10 CALL LEFT1A(TIME)
1442.      WRITE(6,20) TIME
1443.      20 FORMAT (1X,' THE TIME LEFT IS NOW ',F9.5,' SEC.')
1444.      RETURN
1445.      30 WRITE (6,40)
1446.      WRITE (6,41) (BA(I),I= 1,M)
1447.      WRITE (6,49) (BE(I),I=1,N)
1448.      IF (KNU.EC.0) GO TO 35
1449.      WRITE (6,42) ((G1(I,J),I= 1,M),J= 1,KNU)
1450.      35 IF (KMU.EC.0) RETURN
1451.      WRITE (6,43) ((G2(I,J),I= 1,N),J= 1,KMU)
1452.      RETURN
1453.      40 FORMAT (1X,' LINEAR CONSTRAINTS.')
1454.      41 FORMAT (1X,' G1= ',7F13.5)
1455.      49 FORMAT (1X,' BE= ',7F13.5)
1456.      42 FORMAT (1X,' G2= ',7F13.5)
1457.      60 IF (MODE-4) 65,70,100
1458.      65 WRITE (6,50) (KINBAS(I),I=1,NCOL)
1459.      WRITE (6,55) (JH(I),I= 1,NRCW)
1460.      50 FORMAT (1X,' KINEAS= ',20I4)
1461.      55 FORMAT (1X,' JH= ',20I4)
1462.      RETURN
1463.      70 WRITE (6,71) KNU,KMU
1464.      WRITE (6,152) (NUH(I),I=1,IH)
1465.      WRITE (6,151) (MUH(I),I=1,IH)
1466.      WRITE (6,74) (XX(I),I=1,IH)
1467.      WRITE (6,75) (PI(I),I=1,IH)

```

```

1968.      71 FORMAT (//,' SUPEREASIC INFO: KNU=',I3,' KMU= ',I3)
1969.      74 FORMAT ('      T      =',9F12.6)
1970.      75 FORMAT (' LAMBDA=',9F12.6)
1971.      RETURN
1972.    100 IF (MODE - 6) 101,140,140
1973.    101 WRITE (6,120) KNU,KMU
1974.      IF (KMU.EQ.0) GO TO 105
1975.      WRITE (6,121) (BF1(I),I=1,KMU)
1976.      WRITE (6,122) ((D1(I,J),J=1,KMU),I=1,KMU)
1977.      WRITE (6,123) (E1(I),I=1,KMU)
1978.      IF (KNU.EQ.0) GO TO 105
1979.      WRITE (6,124) ((F1(I,J),J=1,KNU),I=1,KMU)
1980.    105 IF (KNU.EQ.0) RETURN
1981.      WRITE (6,125) (BF2(I),I=1,KNU)
1982.      WRITE (6,126) (E2(I),I=1,KNU)
1983.      IF (KMU.EQ.0) RETURN
1984.      WRITE (6,127) ((C2(I,J),J=1,KMU),I=1,KNU)
1985.      WRITE (6,128) ((F2(I,J),J=1,KMU),I=1,KNU)
1986.      RETURN
1987.    120 FORMAT (1X,'BL CONSTRAINTS, KNU,KMU:',I2,I3)
1988.    121 FORMAT (1X,' BF1= ',F14.6)
1989.    122 FORMAT (1X,' D1= ',F14.6)
1990.    123 FORMAT (1X,' E1= ',F14.6)
1991.    124 FORMAT (1X,' F1= ',F14.6)
1992.    125 FORMAT (1X,' BF2= ',F14.6)
1993.    126 FORMAT (1X,' E2= ',F14.6)
1994.    127 FORMAT (1X,' C2= ',F14.6)
1995.    128 FORMAT (1X,' F2= ',F14.6)
1996.    140 WRITE (6,150) PD,KMU,KNU,INEQ
1997.      WRITE (6,151) (MUH(I),I= 1,IH)
1998.      WRITE (6,152) (NUH(I),I= 1,IH)
1999.    150 FORMAT (//,1X,' ENDPPOINT FINDER DATA:PD,KMU,KNU,INEQ= ',4I3)
2000.    151 FORMAT (1X,'MUH(I)= ',20I3)
2001.    152 FORMAT (1X,'NUH(I)= ',20I3)
2002.      WRITE (6,154) (JH(I),I=1,M)
2003.      WRITE (6,155) (DIGMA(I),I=1,N)
2004.    154 FORMAT (1X,'JH= ',15I4)
2005.    155 FORMAT (1X,'DIGMA= ',15I4)
2006.      WRITE (6,153) (XX(I),I=1,NM)
2007.      WRITE (6,156) (PI(I),I=1,NM)
2008.    153 FORMAT (1X,' XX= ',9F13.6)
2009.    156 FORMAT (1X,' PI= ',9F13.6)
2010.      RETURN
2011.      END

```

III. The HRA (Homotopy Retraction Algorithm) Code for Solving Equilibrium Problems

III.1. Revisions to the Original Code

The homotopy retraction algorithm is described in detail in Chapter IV of the report [2]. However, since the completion of that work the author has changed the HRA code to improve the domain of convergence of the code with respect to the user supplied initial utility levels. With these changes the code successfully finds equilibrium points for problems and starting points which resulted in failure for the previous version of the HRA. Also, solution times have been as fast, or faster, than before. This version of the code has not been tested on a sufficient number of test problems but the advantages over the old code seem to be so great that we should describe the code in its present form.

By Lemma II.2.4, solving the equilibrium problem is equivalent to solving for

$$w \equiv (\pi, \lambda, \zeta^i (i \in \underline{m}), p; s, t, z^i (i \in \underline{m}), p)$$

which satisfies

$$f(w) = 0, \quad w \in D, \quad (5)$$

where

$$f(w) = \begin{pmatrix} \pi(w)w^1 - \lambda_1(w)(t_1(w) + v_1) \\ \vdots \\ \pi(w)w^m - \lambda_m(w)(t_m(w) + v_m) \end{pmatrix}$$

$(\pi(w))$ refers to the first m components of w , etc., and

$$D \equiv \{w | \gamma^i Z^i - t_i = v_i \quad i \in \underline{m}\}$$

$$\sum_{i=1}^m B^i Z^i + pe + s = \sum_{i=1}^m w^i$$

$$\pi B^i - \lambda_i \gamma^i - \zeta^i = 0, \quad i \in \underline{m}$$

$$\pi e - p = 1$$

$$\pi s = 0$$

$$\zeta^i z^i = 0, \quad i \in \underline{m}$$

$$w \geq 0 \quad \} .$$

The original homotopy which is used to define a path to the solution of (1.2) is

$$F(w, \theta) = \theta f(w) - (1 - \theta)(\lambda(w) - \lambda^0)$$

where λ^0 is determined by the solution to the auxilliary linear program (2). As part of the modification to HRA we changed the deformation above to

$$F(w, \theta) = \theta f(w) - (1 - \theta)(t(w) - t^0) .$$

Here $t(w)$ refers to the slack variables of the first m rows, where m is the number of consumers in the economy. By the properties of the auxiliary linear program, it will always be true that t^0 , the value of $t(w)$ at the optimum, is identically zero. This is because one can never achieve a higher level of exports p by allowing consumers to have a greater utility level than v_i . Thus, we can simplify the definition of the homotopy to $F: DX[0,1] \rightarrow R^m$,

$$F(w, \theta) = \theta f(w) - (1 - \theta)t(w) . \quad (6)$$

A convergence theorem using the path defined by (6) can be proved even though we will not do so here.

Other amendments to the algorithm include a procedure for altering the initial utility levels v_i , $i = 1, \dots, m$, if the processing of the auxiliary linear program indicates that the solution is unsatisfactory. There are three factors which can cause the program to change the initial utility levels:

1. If the l.p. is infeasible, then v_i is too large. They will all be reduced.
2. If some $\lambda_i^0 = 0$ initially, it indicates that v_i can be increased without changing the optimal value of the objective function. For each i for which this is true we increase v_i until $\lambda_i > 0$.

3. If $f_i(w) \leq 0$ for some i , the initial boundary conditions are not satisfied. By reducing v_i for such indices i , we can increase f_i until it is positive.

When we say to increase v_i or decrease v_i , we mean that the following procedure is followed:

a) After each solution of the auxiliary LP, a scalar TI is incremented by one starting with $TI = 5$.

b) If any of the three conditions above occurs, to reduce v_i we let

$$v_i = v_i * (1 - 1/TI) ,$$

to increase v_i we let

$$v_i = v_i * (1 + 1/TI) .$$

c) When TI reaches 20 and the solution to the LP is still unsatisfactory, the program terminates.

III.2. Input Requirements

The form of the input data for the HRA is exactly identical to the form for the BCA. See Section II.1 for a description.

III.3. Main Program

The main program consists of 390 source statements which perform many of the same functions as the main program for the BCA. The parameters are initialized and possibly altered by reading the name-list PARM1. The subroutines of LPM1 are called to input and solve the auxiliary linear program. Tests 1 and 2 of Section 1 are performed. The data is read so the C matrix can be constructed. Then the budget surpluses f_i are calculated and test 3 is performed. The rest of the main program is essentially identical to that of the BCA except that different decisions are made after the ENDPNT subroutine has been called to determine which is the next cell (see Figure IV.3.1, Elken [1977b]).

Restrictions Relevant to the Use of HRA

1. The number of consumers (IH) must be less than or equal to 10.
2. The matrix D must not have more than 350 rows or 400 columns or more than 4000 non-zero elements.

Of course, if one must solve a problem larger than these dimensions allow, more core must be allocated and the dimensions of the appropriate variables must be changed in every subroutine in which they appear.

Before describing the other subroutines of HRA we will define the variables which differ from those in BCA as described in Section II.2.

Variable Glossary

In HRA we refer to one more variable from the blank common of LPMI.

$$\text{MSTAT} = \begin{cases} \text{'N'} & \text{if the auxiliary linear program} \\ & \text{is infeasible} \\ \text{'O'} & \text{if the current solution is optimal} \\ \text{'U'} & \text{if the LP is unbounded.} \end{cases}$$

We use MSTAT to check if the current auxiliary linear program is infeasible.

The common blocks LP1, BLCST, BLCST2, LNCONS, INDx1, and INDx2 are identical in both HRA and BCA. COMMON/INT/ contains the variable KEND, the number of times the ENDPNT subroutine is called.

We include this variable to pass its value from the main program to ENDPNT so that we can test to see whether this is the first call to ENDPNT or not. In addition to those variables described in Section II.2, COMMON/SCAL/ contains the variable

IHP1 = IH + 1 the number of consumers plus one. This is the dimension of the consumer space plus one for the homotopy parameter.

COMMON/DIM/

$$\text{ITAIL} = \begin{cases} 1, & \text{if the Newton tail routine is in effect,} \\ 0, & \text{otherwise} \end{cases}$$

The description of the algorithm in Chapter IV of Elken [2], shows when ITAIL is set equal to 1.

The rest of the common blocks are identical to those in BCA.

III.4. Subroutines of HRA

We will describe only the differences between the subroutines of HRA and those of BCA.

- 1) Subroutine BLCONS: Calculates the coefficients of the bilinear functionals the budget surpluses so that $f(w)$, (6), can be evaluated in terms of the superbasic variables (see VI (4, 9) of Elken [2]). This subroutine is substantially the same as BLCONS in BCA except that in this case all the bilinear functionals are computed rather than the first DD, the number of bilinear constraints which are currently satisfied.

Subroutines called: UPAKC

- 2) SUBROUTINE FINDP (PD1, IS, P): same as in BCA. See Section II.3.
- 3) SUBROUTINE PIVOT (SS, RR), SUBROUTINE UPAKC (II), SUBROUTINE BSCNG (S, R), SUBROUTINE SUPERB (KEY, PK1, IS, (PD2, JS), and SUBROUTINE RECALC: same as in BCA, see Section II.3.
- 4) SUBROUTINE ENDPNT (JS, PD1, IS, NET): This routine implements the path-following algorithm described in Section IV.2 of [2]. The reader should consult that work to be able to understand this subroutine. Here we will describe some details which were not mentioned in Section II.3.

The independent variables (superbasics) are placed in a vector X of dimension $IH + 1 = IHP1$. The extra dimension is for the homotopy parameter, θ . Theta always occupies the last position in X , $X(IHP1)$.

From computational experience, we know that the curve $F^{-1}(0)$ is highly nonlinear in the last variable when the algorithm begins. That is, for the first tangential approximation in the first cell, θ increases very quickly, initially, but quite slowly thereafter. Remember, θ increases from 0 to 1 as the algorithm progresses, although some decreases are possible. To remedy the poor guess at θ which would result from following the initial tangent all the way to the opposite boundary of the cell, on the first iteration we move only one quarter of the distance to the opposite boundary. This rather ad hoc procedure seems to work quite well. After this initial step, the algorithm described in the work cited is followed precisely. The first tangent of the first cell is characterized by the fact that $KEND = 1$ ($KEND$ is the number of cells traversed) and $INFL = 0$ ($INFL$ is set equal to 1 after the first steplength is reduced).

In this program, the integer variable $MFLAG$ has a different meaning than in the BCA code

$MFLAG =$	{	0 when the tangent to the curve is being computed, 1 when a "hyperplane subproblem" is being solved, 2 when an endpoint in the opposite facet is being solved for, and 3 when the equilibrium point appears to be in the current cell. Newton's method will be implemented as a tail routine on $F(X) = 0$.
-----------	---	--

SUBROUTINES CALLED:

QUADS, DSENT, CONCHK, GTN, FTN, DERIVG, DERIV, NORM, DECOMP, SOLVE.

Subroutines QUADS (U, IMAG, ALPHA, BETA) and DSENT (ALPHA, PNORM) are the same as in BCA.

SUBROUTINE GTN (IFL, Q, U, F) evaluates the homotopy $F(w, \theta)$.

When one writes this function in terms of the superbasic variables one has

$$\text{THETA} = X(\text{IHP1})$$

$$Q(X) = \text{THETA} * f(X)$$

$$-(1 - \text{THETA}) * \begin{pmatrix} \sum_{I=1}^{\text{KNU}} G1(\text{MUH}(J), I) * X(\text{KMU} + I) + \text{BA}(\text{MUH}(J)), & J = 1, \dots, \text{KMU} \\ X(\text{KMU} + J), & J = 1, \dots, \text{KNU} \end{pmatrix}$$

The values of the variables of t indexed by MU (the first KMU in the vector above) are saved in the vector $\text{BLAM}(10)$. These values are used in the subroutine DERIVG to be described later.

The last component of Q contains the value of a functional determined by the value of IFL (MFLAG in subroutine ENDPNT). If $\text{IFL} = 1$, then

$$Q(\text{IHP1}) = \sum_{I=1}^{\text{IH}} \text{ACC}(2, I) * (X(I) - \text{ACC}(1, I)) .$$

If IFL = 2, then

$$Q(IHPl) = \begin{cases} \sum_{I=1}^{KNU} G1(JJ, I) * X(KMU + I) + BA(JJ) , & \text{if } MPD = 1 , \\ \sum_{I=1}^{KMU} G2(JJ, I) * X(I) + BB(JJ) , & \text{if } MPD = -1 . \end{cases}$$

The subroutine FTN is called to evaluate $f(X)$.

SUBROUTINE DERIVG (IFL, U, G, F): calculates the jacobian of the function evaluated by GTN. The jacobian is of the form

$$DG(X) = \left[\left[THETA * Df(X) - (1. - THETA) \left(\frac{0}{0} \mid \frac{G1_{MUH, \cdot}}{I} \right) \right] f(X) + \left(\frac{BLAM}{X_{NU}} \right) \right]$$

SUBROUTINE CONCHK (GMIN, GKMIN, MP2): this subroutine evaluates the basic variables in both the primal and dual systems, saves the value of the minimum in GMIN and points to which variable it is with (KGMIN, MP2). From the theory of the homotopy retraction algorithm, it is known that the first IH primal and dual variables cannot become negative so we ignore these variables when searching for the minimum. Also, the primal and dual variables corresponding to the objective function are of no importance to the algorithm, so we do not compare these values when searching for the minimum.

SUBROUTINE FTN(F, Y): evaluates the bilinear budget surpluses and stores them in F. This subroutine is identical to that in the BCA code except that no additional functionals are evaluated. That task is performed by GTN as described above.

SUBROUTINE DERIV: evaluates the jacobian of the function described in FTN. This is identical to the subroutine by the same name in the BCA code except, again, it is simpler because no rows need be appended because of an additional functional.

SUBROUTINES NORM (Y, S1, NY, DECOMP (NN, A, UL), SOLVE (NN, UL, B, X), SING (IWHY), DEBUG (MODE): are identical to the subroutines with the same names in BCA.

III.5. Sample Problems

Below we give the output of the HRA program for the same two problems (MAS-Colell and Whisman) presented in Section II.4. The form of the input is identical for HRA and BCA so we will not repeat the input decks here.

The output is very similar. The reader will note that the subproblem solved by ENDPNT is $(IH + 1)$ -dimensional with HRA while the dimension grew from 1 to IH for the BCA. On these two problems, though, the HRA code required fewer calls to ENDPNT before equilibrium is reached. This behavior is typical with larger problems as well.

1598.
1599.
1600.
1601.
1602.
1603.
1604.
1605.
1606.
1607.
1608.
1609.
1610.
1611.
1612.
1613.
1614.
1615.
1616.
1617.
1618.
1619.
1620.
1621.
1622.
1623.
1624.
1625.
1626.
1627.
1628.
1629.
1630.
1631.
1632.

C.0115164
C.0
WE MOVE A DISTANCE
THE CURVE HIT A 1 TYPE CONSTRAINT, NUMBER 5

0.5868394
0.5916951
C.394011
0.0339351
0.6122210
ALONG THE APPROX.

MFLAG= 2
ITERATION :
NORM(F(X))= 0.065775 F= 0.048328 -0.044450 -0.003978
ITERATION :
NORM(F(X))= 0.000313 F= 0.000128 -0.000255 0.000128
ITERATION :
NORM(F(X))= 0.000000 F= 0.000000 -0.000000 0.000000
ITERATION :
NORM(F(X))= 0.000000 F= 0.000000 -0.000000 0.000000
ITERATION :
NORM(F(X))= 0.000000 F= 0.000000 -0.000000 0.000000
AFTER 3 NEWTON ITERATIONS,

AFTER 2 ENDPNT CALLS, WE HAVE EQUILIBRIUM.

TOTALS: SCALAR FUNCTION CALLS= 29
JACOBIAN EVALUATION= 7

THE TIME LEFT IS NOW 14.876805 SEC.
UTILITY LEVELS: 1.5773504 1.1547005
PRICES: 0.633975 0.866025 0.236603 4.2264982
WAS-COL

JH(I)	VALUE	ROW NAMES	PI(I)	RHS
ACT1	1.57735041	UTIL1	0.63397455	-0.90000000
ACT2	1.15470049	UTIL2	0.86602545	-0.95000000
ACT3	4.22649816	UTIL3	0.23660254	-3.92000000
UTIL1	0.67735041	G0001	0.73205090	3.00000000
WEXP	-0.00000023	G0002	0.26794910	3.00000000
OBJ	0.000000023	OBJ	-1.00000000	0.0

AD-A061 380

STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB

F/G 5/3

BCA AND HRA: TWO PROGRAMS FOR COMPUTING ECONOMIC EQUILIBRIA.(U)

AUG 78 T ELKEN

N00014-75-C-0267

UNCLASSIFIED

SOL-78-17

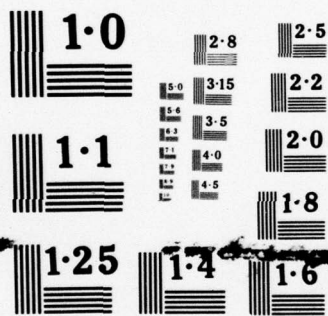
NL

2 OF 2
ADA
081380



END
DATE
FILMED

2 -79
DDC



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

1J49.
1J50.
1J51.
1J52.
1J53.
1J54.
1J55.
1J56.
1J57.
1J58.
1J59.
1J60.
1J61.
1J62.
1J63.
1J64.
1J65.
1J66.
1J67.
1J68.
1J69.
1J70.
1J71.
1J72.
1J73.
1J74.
1J75.
1J76.
1J77.
1J78.
1J79.
1J80.
1J81.
1J82.
1J83.
1J84.
1J85.
1J86.
1J87.
1J88.
1J89.
1J90.
1J91.
1J92.
1J93.
1J94.
1J95.
1J96.
1J97.
1J98.
1J99.
1100.
1101.
1102.
1103.
1104.
1105.
1106.
1107.
1108.

MEXP
CAAC2
GUDD3
OBJ

2.58000000
0.03000000
4.34000000
-2.58000000

GOOD1
GOOD2
GOOD3
OBJ

5.00000000
11.00000000
9.00000000
0.0

3.50000000
0.50000000
-0.00000000
-1.00000000

THE TIME LEFT IS NOW 14.886756 SEC.

STRUCTURAL COLUMNS OF MATRIX.

COLUMNS	1	THROUGH	4
-0.4800	0.0	0.0	0.0
0.0	-0.3200	0.0	0.0
0.0	0.0	-5.9000	0.0
0.0	0.0	0.0	-0.9500
2.0000	1.0000	1.0000	1.0000
2.0000	3.0000	5.0000	1.0000
1.0000	3.0000	4.0000	1.0000
0.0	0.0	0.0	0.0

SMAL= 1.1718749 THETA= 0.5395683
WE RETRACT CNIC THE 1 TH BOUNDARY.
XO= 0.000000 1.093750 0.441406 0.941406
INITIAL BUDGET SURPLUSES :
1.280000 1.200000 0.050000 0.050000

THE 1TH ENDPT CALL:
THE CURRENT QUADRATIC APPROXIMATION TO THE CURVE IS:
F(X)
A2(*U)+
A3

1.2800001 0.2396748 0.0
1.2000000 0.4723171 0.0
0.0500002 0.0583558 0.0
0.0500000 0.0401557 0.0
0.0 0.2444575 0.5395683

WE MOVE A DISTANCE 1.016266 ALONG THE APPROX.
THE CURVE HIT A 1 TYPE CONSTRAINT, NUMBER 5

MFLAG= 2
ITERATION : C X= 0.85333 0.48000 0.10000 0.05000
NORM(F(X))= 0.000000 F= 0.00000 0.00000 0.00000 0.00000
ITERATION : 1 X= 0.85333 0.48000 0.10000 0.05000
NORM(F(X))= 0.0 F= 0.0 0.0 0.0 0.0
AFTER 1 NEWTON ITERATIONS.

AFTER 1 ENDPT CALLS, WE HAVE EQUILIBRIUM.

TOTALS: SCALAR FUNCTION CALLS= 16
JACOBIAN EVALUATIONS= 2

THE TIME LEFT IS NOW 14.865749 SEC.
UTILITY LEVELS: 1.3333334 0.8000000 6.0000000 1.0000000
PRICES: 1.500000 2.500000 0.500000 1.000000

WHISMAN

1109.
1110.
1111.
1112.
1113.
1114.
1115.
1116.
1117.
1118.
1119.
1120.
1495.
1496.
1497.

JH(1)
C1AC1
C2AC2
C3AC1
C4AC1
MEXP
C4AC2
GUDD3
GBJ

VALUE
1.33333337
C.80000001
C.000000028
C.733333314
-0.000000028
C.266666687
3.400000015
C.000000028

ROW NAMES
UTIL1
UTIL2
UTIL3
UTIL4
GUDD1
GUDD2
GUDD3
GBJ

PI(1)
1.500000000
2.500000000
0.500000000
1.000000000
0.500000000
0.500000000
-0.500000000
-1.000000000

RHS
-0.480000000
-0.320000000
-5.900000000
-0.950000000
5.000000000
11.000000000
0.000000000
0.0

III.6. HRA Source Listing

Below is a listing of HRA without the subroutines which comprise the linear programming code LPML.


```

1.      IMPLICIT REAL*8 (A-H,O-Z)
2.      REAL*4 ZTOLZE,ZTOLPV,ZSCALE,ZTETA,ZTOLRP,ZTOLRI
3.      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P,QN
4.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
5.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PLN,LC(20),IC(800)
6.      DOUBLE PRECISION E(8000)
7.      REAL A(4000),C(300),CMIN,COND,ERMAX,SUMINF
8.
9.      C
10.     COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
11.     1A,S,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
12.     2NTEMP(20),KINP,ITIM,ITIM,ITINV,ITINV,MSTAT,I0BJ,I0WP,IVIN,IVOUT,
13.     3ITCNT,INVERO,ITRLIM,IFFEZ,JCOLP,NRCW,NCOL,NELEM,NETA,NLELEM,NLETA,
14.     4NGLECM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
15.     5LA(1302),LE(2002),PUN(8),
16.     6IPUNC,NDEGI,NDUAL,NIP1W,IFEAS,IFCRSH
17.     COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
18.     COMMON IA(4000),IE(8000)
19.
20.     C
21.     DATA QN/'N' /
22.     COMMON/LP1/PI(1302),XX(1302)
23.     COMMON/RLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
24.     COMMON/ELCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
25.     COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
26.     COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
27.     COMMON/INDX2/ JH,DIGMA,KINEAS,IDBAS
28.     COMMON/SCAL/ ST,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
29.     COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
30.     COMMON/INT/ IPS(30),KDET,KCUNT,ISING,KEND
31.     COMMON/TOLER/ TOLFZ,TOLBD,TOLCV,THETA,STPMX,STPRD
32.     DIMENSION V3(10),F(10),SHR(10)
33.     NAMELIST/PA1/TOLFZ,TOLBD,TOLCV,ICNTRL,IECHO,
34.     1STPMX,STPRD,ICECHO,IH,L,IPROD,KOUTE,ITLIME
35.
36.     C
37.     C
38.     C
39.     C
40.     C
41.     C
42.     C
43.     C
44.     C
45.     C
46.     C
47.     C
48.     C
49.     C
50.     C
51.     C
52.     C
53.     C
54.     C
55.     C
56.     C
57.     C
58.     C
59.     C
60.     C
61.     C
62.     C
63.     C
64.     C
65.     C
66.     C
67.     C
68.     C
69.     C
70.     C
71.     C
72.     C
73.     C
74.     C
75.     C
76.     C
77.     C
78.     C
79.     C
80.     C
81.     C
82.     C
83.     C
84.     C
85.     C
86.     C
87.     C
88.     C
89.     C
90.     C
91.     C
92.     C
93.     C
94.     C
95.     C
96.     C
97.     C
98.     C
99.     C
100.    C
101.    C
102.    C
103.    C
104.    C
105.    C
106.    C
107.    C
108.    C
109.    C
110.    C
111.    C
112.    C
113.    C
114.    C
115.    C
116.    C
117.    C
118.    C
119.    C
120.    C
121.    C
122.    C
123.    C
124.    C
125.    C
126.    C
127.    C
128.    C
129.    C
130.    C
131.    C
132.    C
133.    C
134.    C
135.    C
136.    C
137.    C
138.    C
139.    C
140.    C
141.    C
142.    C
143.    C
144.    C
145.    C
146.    C
147.    C
148.    C
149.    C
150.    C
151.    C
152.    C
153.    C
154.    C
155.    C
156.    C
157.    C
158.    C
159.    C
160.    C
161.    C
162.    C
163.    C
164.    C
165.    C
166.    C
167.    C
168.    C
169.    C
170.    C
171.    C
172.    C
173.    C
174.    C
175.    C
176.    C
177.    C
178.    C
179.    C
180.    C
181.    C
182.    C
183.    C
184.    C
185.    C
186.    C
187.    C
188.    C
189.    C
190.    C
191.    C
192.    C
193.    C
194.    C
195.    C
196.    C
197.    C
198.    C
199.    C
200.    C
201.    C
202.    C
203.    C
204.    C
205.    C
206.    C
207.    C
208.    C
209.    C
210.    C
211.    C
212.    C
213.    C
214.    C
215.    C
216.    C
217.    C
218.    C
219.    C
220.    C
221.    C
222.    C
223.    C
224.    C
225.    C
226.    C
227.    C
228.    C
229.    C
230.    C
231.    C
232.    C
233.    C
234.    C
235.    C
236.    C
237.    C
238.    C
239.    C
240.    C
241.    C
242.    C
243.    C
244.    C
245.    C
246.    C
247.    C
248.    C
249.    C
250.    C
251.    C
252.    C
253.    C
254.    C
255.    C
256.    C
257.    C
258.    C
259.    C
260.    C
261.    C
262.    C
263.    C
264.    C
265.    C
266.    C
267.    C
268.    C
269.    C
270.    C
271.    C
272.    C
273.    C
274.    C
275.    C
276.    C
277.    C
278.    C
279.    C
280.    C
281.    C
282.    C
283.    C
284.    C
285.    C
286.    C
287.    C
288.    C
289.    C
290.    C
291.    C
292.    C
293.    C
294.    C
295.    C
296.    C
297.    C
298.    C
299.    C
300.    C
301.    C
302.    C
303.    C
304.    C
305.    C
306.    C
307.    C
308.    C
309.    C
310.    C
311.    C
312.    C
313.    C
314.    C
315.    C
316.    C
317.    C
318.    C
319.    C
320.    C
321.    C
322.    C
323.    C
324.    C
325.    C
326.    C
327.    C
328.    C
329.    C
330.    C
331.    C
332.    C
333.    C
334.    C
335.    C
336.    C
337.    C
338.    C
339.    C
340.    C
341.    C
342.    C
343.    C
344.    C
345.    C
346.    C
347.    C
348.    C
349.    C
350.    C
351.    C
352.    C
353.    C
354.    C
355.    C
356.    C
357.    C
358.    C
359.    C
360.    C
361.    C
362.    C
363.    C
364.    C
365.    C
366.    C
367.    C
368.    C
369.    C
370.    C
371.    C
372.    C
373.    C
374.    C
375.    C
376.    C
377.    C
378.    C
379.    C
380.    C
381.    C
382.    C
383.    C
384.    C
385.    C
386.    C
387.    C
388.    C
389.    C
390.    C
391.    C
392.    C
393.    C
394.    C
395.    C
396.    C
397.    C
398.    C
399.    C
400.    C
401.    C
402.    C
403.    C
404.    C
405.    C
406.    C
407.    C
408.    C
409.    C
410.    C
411.    C
412.    C
413.    C
414.    C
415.    C
416.    C
417.    C
418.    C
419.    C
420.    C
421.    C
422.    C
423.    C
424.    C
425.    C
426.    C
427.    C
428.    C
429.    C
430.    C
431.    C
432.    C
433.    C
434.    C
435.    C
436.    C
437.    C
438.    C
439.    C
440.    C
441.    C
442.    C
443.    C
444.    C
445.    C
446.    C
447.    C
448.    C
449.    C
450.    C
451.    C
452.    C
453.    C
454.    C
455.    C
456.    C
457.    C
458.    C
459.    C
460.    C
461.    C
462.    C
463.    C
464.    C
465.    C
466.    C
467.    C
468.    C
469.    C
470.    C
471.    C
472.    C
473.    C
474.    C
475.    C
476.    C
477.    C
478.    C
479.    C
480.    C
481.    C
482.    C
483.    C
484.    C
485.    C
486.    C
487.    C
488.    C
489.    C
490.    C
491.    C
492.    C
493.    C
494.    C
495.    C
496.    C
497.    C
498.    C
499.    C
500.    C
501.    C
502.    C
503.    C
504.    C
505.    C
506.    C
507.    C
508.    C
509.    C
510.    C
511.    C
512.    C
513.    C
514.    C
515.    C
516.    C
517.    C
518.    C
519.    C
520.    C
521.    C
522.    C
523.    C
524.    C
525.    C
526.    C
527.    C
528.    C
529.    C
530.    C
531.    C
532.    C
533.    C
534.    C
535.    C
536.    C
537.    C
538.    C
539.    C
540.    C
541.    C
542.    C
543.    C
544.    C
545.    C
546.    C
547.    C
548.    C
549.    C
550.    C
551.    C
552.    C
553.    C
554.    C
555.    C
556.    C
557.    C
558.    C
559.    C
560.    C
561.    C
562.    C
563.    C
564.    C
565.    C
566.    C
567.    C
568.    C
569.    C
570.    C
571.    C
572.    C
573.    C
574.    C
575.    C
576.    C
577.    C
578.    C
579.    C
580.    C
581.    C
582.    C
583.    C
584.    C
585.    C
586.    C
587.    C
588.    C
589.    C
590.    C
591.    C
592.    C
593.    C
594.    C
595.    C
596.    C
597.    C
598.    C
599.    C
600.    C
601.    C
602.    C
603.    C
604.    C
605.    C
606.    C
607.    C
608.    C
609.    C
610.    C
611.    C
612.    C
613.    C
614.    C
615.    C
616.    C
617.    C
618.    C
619.    C
620.    C
621.    C
622.    C
623.    C
624.    C
625.    C
626.    C
627.    C
628.    C
629.    C
630.    C
631.    C
632.    C
633.    C
634.    C
635.    C
636.    C
637.    C
638.    C
639.    C
640.    C
641.    C
642.    C
643.    C
644.    C
645.    C
646.    C
647.    C
648.    C
649.    C
650.    C
651.    C
652.    C
653.    C
654.    C
655.    C
656.    C
657.    C
658.    C
659.    C
660.    C
661.    C
662.    C
663.    C
664.    C
665.    C
666.    C
667.    C
668.    C
669.    C
670.    C
671.    C
672.    C
673.    C
674.    C
675.    C
676.    C
677.    C
678.    C
679.    C
680.    C
681.    C
682.    C
683.    C
684.    C
685.    C
686.    C
687.    C
688.    C
689.    C
690.    C
691.    C
692.    C
693.    C
694.    C
695.    C
696.    C
697.    C
698.    C
699.    C
700.    C
701.    C
702.    C
703.    C
704.    C
705.    C
706.    C
707.    C
708.    C
709.    C
710.    C
711.    C
712.    C
713.    C
714.    C
715.    C
716.    C
717.    C
718.    C
719.    C
720.    C
721.    C
722.    C
723.    C
724.    C
725.    C
726.    C
727.    C
728.    C
729.    C
730.    C
731.    C
732.    C
733.    C
734.    C
735.    C
736.    C
737.    C
738.    C
739.    C
740.    C
741.    C
742.    C
743.    C
744.    C
745.    C
746.    C
747.    C
748.    C
749.    C
750.    C
751.    C
752.    C
753.    C
754.    C
755.    C
756.    C
757.    C
758.    C
759.    C
760.    C
761.    C
762.    C
763.    C
764.    C
765.    C
766.    C
767.    C
768.    C
769.    C
770.    C
771.    C
772.    C
773.    C
774.    C
775.    C
776.    C
777.    C
778.    C
779.    C
780.    C
781.    C
782.    C
783.    C
784.    C
785.    C
786.    C
787.    C
788.    C
789.    C
790.    C
791.    C
792.    C
793.    C
794.    C
795.    C
796.    C
797.    C
798.    C
799.    C
800.    C
801.    C
802.    C
803.    C
804.    C
805.    C
806.    C
807.    C
808.    C
809.    C
810.    C
811.    C
812.    C
813.    C
814.    C
815.    C
816.    C
817.    C
818.    C
819.    C
820.    C
821.    C
822.    C
823.    C
824.    C
825.    C
826.    C
827.    C
828.    C
829.    C
830.    C
831.    C
832.    C
833.    C
834.    C
835.    C
836.    C
837.    C
838.    C
839.    C
840.    C
841.    C
842.    C
843.    C
844.    C
845.    C
846.    C
847.    C
848.    C
849.    C
850.    C
851.    C
852.    C
853.    C
854.    C
855.    C
856.    C
857.    C
858.    C
859.    C
860.    C
861.    C
862.    C
863.    C
864.    C
865.    C
866.    C
867.    C
868.    C
869.    C
870.    C
871.    C
872.    C
873.    C
874.    C
875.    C
876.    C
877.    C
878.    C
879.    C
880.    C
881.    C
882.    C
883.    C
884.    C
885.    C
886.    C
887.    C
888.    C
889.    C
890.    C
891.    C
892.    C
893.    C
894.    C
895.    C
896.    C
897.    C
898.    C
899.    C
900.    C
901.    C
902.    C
903.    C
904.    C
905.    C
906.    C
907.    C
908.    C
909.    C
910.    C
911.    C
912.    C
913.    C
914.    C
915.    C
916.    C
917.    C
918.    C
919.    C
920.    C
921.    C
922.    C
923.    C
924.    C
925.    C
926.    C
927.    C
928.    C
929.    C
930.    C
931.    C
932.    C
933.    C
934.    C
935.    C
936.    C
937.    C
938.    C
939.    C
940.    C
941.    C
942.    C
943.    C
944.    C
945.    C
946.    C
947.    C
948.    C
949.    C
950.    C
951.    C
952.    C
953.    C
954.    C
955.    C
956.    C
957.    C
958.    C
959.    C
960.    C
961.    C
962.    C
963.    C
964.    C
965.    C
966.    C
967.    C
968.    C
969.    C
970.    C
971.    C
972.    C
973.    C
974.    C
975.    C
976.    C
977.    C
978.    C
979.    C
980.    C
981.    C
982.    C
983.    C
984.    C
985.    C
986.    C
987.    C
988.    C
989.    C
990.    C
991.    C
992.    C
993.    C
994.    C
995.    C
996.    C
997.    C
998.    C
999.    C
1000.   C

```



```

51.      IF (ICNTRL) 10,40,40
52. 10 CALL INPUT(IACT)
53.      IF (IACT) 30, 20, 30
54. 20 IF (TECH0.F0.0) GO TO 29
55.      MUR= NROW
56.      WRITE (6,21)
57. 21 FORMAT(//////, '          STRUCTURAL COLUMNS OF MATRIX.')
58. 22 LB= MUR+1
59.      MUR= LB+ 14
60.      K= 0
61.      DO 25 J= LB, MUR
62.          IF (J.GT.NCOL) GO TO 26
63.          K= K+ 1
64.          CALL UNPACK(J)
65.          DO 24 I= 1, NROW
66.              G1(I,K)= Y(I)
67. 24 CONTINUE
68. 25 CONTINUE
69. 26 WRITE (6,27) LB, MUR
70. 27 FORMAT (//////, ' COLUMNS ', I4, ' THROUGH ', I4)
71.      DO 23 I=1, NROW
72.          WRITE (6,28) (G1(I,J), J=1, K)
73. 28 FORMAT (1X, 15F8.3 )
74. 23 CONTINUE
75.      IF (MUR .LT. NCOL) GO TO 22
76. 29 CALL NORMAL
77.      CALL UNPAVL(0)
77.1      GO TO 10
77.2
77.3 30 STOP
77.4 40 CALL INPUT(IACT)
78.      CALL DEBPG(1)
78.1      MFLAG= 0
78.2      ITRY= 0
78.3      TI= 5.
78.4 343 IF (ITRY .LE. 10) GO TO 344
78.5      WRITE (6,342)
78.6 342 FORMAT(' MUST TERMINATE: TOO MANY TRIES AT INITIALIZATION
78.7      ' OF V WITHOUT SUCCESS.')
78.8 344 ITRY= ITRY + 1
78.9      CALL NORMAL
79.      CALL UNPAVL(MFLAG)
79.1      MFLAG= 0
79.2      TI= TI+ 1.
79.3      IF (MSTAT .NE. 0N) GO TO 348
79.4      DO 345 I= 1, IH
79.5          B(I)= (1.0 - 1./TI) * B(I)
79.6 345 GO TO 343
79.7
79.8 348 DO 349 I= 1, IH
79.9      IF (Y(I) .GT. TOL3D) GO TO 349
80.      B(I)= (1.0 + 1./TI) * E(I)
80.1      MFLAG= 1
80.2 349 CONTINUE
80.3      IF (MFLAG .GT. 0) GO TO 343
81. C
81.      CALL DEBPG(1)
81.      CALL SHIFTP(3,4)
81.      KEND= 0
81.      KENDSV= 0
81.      IHL= IH+ L
81.      M= NROW

```

```

92.      NE=NCOL- NROW
93.      NM= N+ M
94.      LI= M- L
95.      MM1= M- 1
96.      ITSINV= 1
97.      IF (IPROD.EQ.0) GO TO 43
98.      READ (5,1030) (SHR(I),I=1,IH)
99.
100. 43 ICOU=0
101.     ISAV= 0
102.     LC(1)= 1
103.     DO 45 K= 1,IH
104.         KPI= K+ 1
105.         READ(KINP,1010) ISHR, LL, KK
106.         IF (ISHR.EQ.1) GO TO 50
107.         IF (ISHR.EQ.-1) GO TO 42
108.         IF (LL.EQ.0) GO TO 58
109.         IF (KK.NE.0) GO TO 47
110.         ICOU= ICOU+ 1
111.         IC(ICOU)= LL
112.         C(ICOU)= B(LL)
113.         GO TO 45
114. 45 DO 46 I= LL, KK
115.         ICOU= ICOU+ 1
116.         IC(ICOU)= I
117.         C(ICOU)= B(I)
118. 46 CONTINUE
119.         GO TO 45
120. 42 IF (KK.NE.0) GO TO 44
121.         ICOU= ICOU+ 1
122.         IC(ICOU)= LL
123.         READ (5,1030) C(ICOU)
124.         GO TO 45
125. 44 ISAV= ICOU+ 1
126.         DO 46 I= LL, KK
127.             ICOU= ICOU+ 1
128.             IC(ICOU)= I
129. 46 CONTINUE
130.         READ(5,1030) (C(J),J=ISAV,ICOU)
131.         GO TO 45
132. 50 IF (KK.NE.0) GO TO 53
133.         ICOU= ICOU+ 1
134.         IC(ICOU)= LL
135.         C(ICOU)= SHR(K)*B(LL)
136.         GO TO 45
137. 53 DO 55 I= LL, KK
138.         ICOU= ICOU+ 1
139.         C(ICOU)= SHR(K)*B(I)
140.         IC(ICOU)= I
141. 55 CONTINUE
142.         GO TO 45
143. 59 LC(KPI)= ICOU+1
144. 68 CONTINUE
145.         IF (ICECHO.EQ.0) GO TO 69
146.         MUR= 0
147.         WRITE (6,61)
148. 61 FORMAT(//////, '          STRUCTURAL COLUMNS OF MATRIX. ')
149.         LB= MUR+1
150.         MUR= LB+ IH - 1
151.         K= 0
152.         DO 65 J= LB, MUR

```

```

152.          IF (J.GT.NCOL) GO TO 66
153.          K= K+ 1
154.          CALL UNPAKE(J)
155.          DO 64 I= 1,NROW
156.             G1(I,K)= Y(I)
157. 64         CONTINUE
158. 65 CONTINUE
159. 66 WRITE (6,67) LB,MUB
160. 67 FORMAT (//////,' COLUMNS ',I4,' THROUGH ',I4)
161.          DO 74 I= 1,NROW
162.             WRITE (6,73) (G1(I,J),J=1,IH)
163. 73 FORMAT (1X,10F10.4)
164. 74 CONTINUE
165. 69 DO 70 I= 1,M
166.            IR= JH(I)
167.            XX(IR)= X(I)
168.            PI(I)= YTEMP(I)
169.            RA(I)= X(I)
170. 70 CONTINUE
171.            MP1= M+ 1
172.            DO 100 J= MP1,NM
173.               IF (KINBAS(J).NE. 0) GO TO 90
174.               DSUM= 0.
175.               LL= LA(J)
176.               KK= LA(J+1) - 1
177.               DO 80 I= LL,KK
178.                  IR= IA(I)
179.                  DE= A(I)
180.                  DPROD= DE*YTEMP(IR)
181.                  DSUM= DSUM + DPROD
182. 80 CONTINUE
183.               PI(J)= DSUM
184.               GO TO 100
185. 90 PI(J)= 0.
186. 100 CONTINUE
187.            K= 0
188.            DO 110 I= 1,NM
189.               IF (KINBAS(I).EQ. 0) GO TO 105
190.               IDBAS(I)= 0
191.               GO TO 110
192. 105 K= K+ 1
193.               SIGMA(K)= 1
194.               IR(K)= PI(I)
195.               IDBAS(I)= K
196.               XX(I)= 0.0
197. 110 CONTINUE
198. C      THIS IS THE BILINEAR PHASE OF THE ALGORITHM. VARIABLES XX(1).....
199. C      XX(M) ARE THE SLACKS. AND PI(1).....,PI(M) ARE THE USUAL PI'S.
200. C      XX(M+1).....,XX(M+N) ARE THE X'S. PI(M+1).....,PI(M+N) ARE THE
201. C      DUAL SLACKS. MORE INITIALIZATIONS.
202. C
203.          LP= 2
204.          ITAIL= 0
205.          PD= 1
206.          JS= 1
207.          DO 120 I=1,IH
208.             NUH(I)= 0
209.             NUH(I)= 0
210.             NU(I)= 0
211.             MU(I)= 0

```



```

212. 120 CONTINUE
213.   KNU= 0
214.   KNU= 0
215.   DO 430 I= 1,IH
216.     IF (KINBAS(I).GT.0) GO TO 425
217.     CALL SUPERB(1,1,1,0,0)
218.     GO TO 430
219. 425   CALL SUPERB(1,-1,1,0,0)
220. 430 CONTINUE
221.     IF (KMU.EQ.0) GO TO 435
222.     DO 433 I= 1,KMU
223. 433   VS(I)= PI(MU(I))
224. 435   IF (KNU.EQ.0) GO TO 440
225.     DO 438 I= 1,KNU
226.       IK= I+ KMU
227. 438   VS(IK)= XX(MU(I))
228.   CALL BLCONS
229. 440   CALL FTN(F,VS)
230.     IO= 1
231.     THETA= 0.0
232.     WRITE (6,449) (F(I),I=1,IH)
233. 449   FORMAT (1X,' INITIAL BUDGET SURPLUSES :',/,9F13.6)
234.     DO 130 I= 1,IH
235.       IF (F(I).GT.0) GO TO 130
236.       B(I)= B(I) + F(I)/PI(I)
237.       MFLAG= 1
238. 130 CONTINUE
239.     IF (MFLAG.GT. 0) GO TO 343
240.     ZFLAG= 0
241.     GO TO 460
242. 450   NTEMP(1)= NTEMP(1)+ JTINV
243.     CALL INVERT
244.     CALL RECALC
245.     ITSINV= 0
246. 460   KEND= KEND+ 1
247.     ITSINV= ITSINV + 1
248.     IF (KEND.GT. ITIME) STOP
249.     WRITE (6,222) KEND
250. 222   FORMAT (/,'-----')
251.     1,/, ' THE ',I4,'TH ENDPNT CALL:')
252.     CALL ENDPNT(JS,PD1,IS,ZFLAG,NET)
253.     IF ( (KEND/3)*3 .NE. KEND ) GO TO 468
254.     CALL DEBUG(4)
255.     CALL DEBUG(1)
256. 468   WRITE (6,470) NET
257. 470   FORMAT (1X,' AFTER ',I3,' NEWTON ITERATIONS,')
258.     IF (ZFLAG.EQ.2) GO TO 920
259.     IF (PD1.EQ.1 .AND. IS.EQ.NM) GO TO 920
260.     IF (PD1) 300,310,310
261. 300   WRITE (6,305) IS
262. 305   FORMAT (1X,' THE ',I3,'TH DUAL VARIABLE WENT TO ZERO.')
263.     GO TO 480
264. 310   WRITE (6,315) IS
265. 315   FORMAT (1X,' THE ',I3,'TH PRIMAL VARIABLE WENT TO ZERO.')
266. 480   IF (PD1.EQ.-1) GO TO 520
267. C
268. C   A PRIMAL VARIABLE WENT TO ZERO. THAT VARIABLE MUST GO TO
269. C   NONBASIC, AND THE INCOMING VARIABLE IS DETERMINED BY FINDING THE
270. C   LARGEST ELEMENT IN THE APPROPRIATE ROW OF G1.

```



```

298.      PD= -1
299.      TROWP= KINBAS(IS)
300.      CALL FINDP(1,IS,JCOLP)
301.      INNAM1= ICNAM(JCOLP,1)
302.      INNAM2= ICNAM(JCOLP,2)
303.      IONAM1= ICNAM(IS,1)
304.      IONAM2= ICNAM(IS,2)
305.      WRITE(6,485) INNAM1,INNAM2,IONAM1,IONAM2
306. 485 FORMAT(1X,' VECIN: ',2A4,' VECOUT: ',2A4)
307.      CALL PIVOT(JCOLP,IS)
308.      CALL UNPACK(JCOLP)
309.      CALL FTRAN(1)
310.      CALL WRETA
311.      CALL BSCNG(JCOLP,IS)
312.      CALL SUPERR(0,-1,JCOLP,1,JCOLP)
313.      CALL BLCONS
314.      JS= IS
315.      IF (ITSINV.GE. INVFRQ) GO TO 450
316.      GO TO 460
317.
318. C      A DUAL VARIABLE WENT TO ZERO. THAT VARIABLE MUST ENTER THE
319. C      BASIS, AND THE LEAVING VARIABLE MUST BE DETERMINED BY FINDING
320. C      THE LARGEST PIVOT ELEMENT IN THE IDBAS(IS)-ROW OF G2.
321. C
322. 520 PD= 1
323.      JCOLP= IS
324.      CALL FINDP(-1,IS,P)
325.      TROWP= KINBAS(P)
326.      INNAM1= ICNAM(IS,1)
327.      INNAM2= ICNAM(IS,2)
328.      IONAM1= ICNAM(P,1)
329.      IONAM2= ICNAM(P,2)
330.      WRITE(6,485) INNAM1,INNAM2,IONAM1,IONAM2
331.      CALL PIVOT(JCOLP,P)
332.      CALL UNPACK(JCOLP)
333.      CALL FTRAN(1)
334.      CALL WRETA
335.      CALL BSCNG(JCOLP,P)
336.      CALL SUPERR(0,1,P,-1,P)
337.      CALL BLCONS
338.      JS= IS
339.      IF (ITSINV.GE. INVFRQ) GO TO 450
340.      GO TO 460
341.
342. C      ANOTHER BILINEAR CONSTRAINT IS SATISFIED. WE INCREASE IH
343. C      AND WE HAVE A POINT IN W(IH).
344. C
345. 920 WRITE(6,930) KEND
346. 930 FORMAT(//,' AFTER',I4,' ENCPNT CALLS, WE HAVE EQUILIBRIUM.')
347.      WRITE(6,920) KFUN,KJAC
348. 320 FORMAT(//,' TOTALS: SCALAR FUNCTION CALLS=',I8,/,11X,' JACOBIAN EVA
349.      ILUATIONS=',I8,/)
350.      CALL DEBUG(1)
351.      DO 935 I= 1,IH
352. 935 VS(I)= -B(I)+ XX(I)
353.      WRITE(6,940) (VS(I),I=1,IH)
354.      WRITE(6,945) (PI(I),I= 1, IH )
355. 940 FORMAT(1X,' UTILITY LEVELS: ',6F15.7)
356. 945 FORMAT(1X,' DUAL MULTIPLIERS: ',9F12.6)
357.      DO 950 I= 1,NROW

```

```

358.          IV= JH(I)
359.          X(I)= XX(IV)
360.          Y(I)= PI(I)
361.      950 CONTINUE
362.          CALL UNRAVL(0)
363.          STOP
364.      1010 FORMAT (10I4)
365.      1030 FORMAT (10F7.4 )
366.          END
367.      SUBROUTINE BLCONS
368.
369.      C      WE ARE GOING TO CALCULATE THE IH ROWS OF COEFFICIENTS
370.      C      FOR THE BILINEAR EQUATIONS AND THE BILINEAR INEQUALITY.
371.      C      THE BASIC MATHEMATICAL STRUCTURE IS THE FOLLOWING:
372.      C
373.      C      BF1 + D1*PI(MU)- DIAG(PI(MU))*(F1*X(NU) + E1) = 0
374.      C      BF2 + D2*PI(MU)- DIAG(X(NU))*(F2*PI(MU) + E2) = 0
375.      C
376.      C      IMPLICIT REAL*8 (A-H,O-Z)
377.      C      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
378.      C      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
379.      C      INTEGER*2 ISTYPE,LA,LE,IA,IE,PLN,LC(20),IC(800)
380.      C      DOUBLE PRECISION E(8000)
381.      C      REAL A(4000),C(800),CMIN,CEND,ERNAX,SUMINF
382.
383.      C      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
384.      C      IA,E,CMIN,CEND,ERNAX,SUMINF,ICNAM(1302,2),NAME(20),
385.      C      2NTRMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
386.      C      3ITCNT,INVERQ,ITRLIM,IFFEZ,JCOLP,NRCW,NCOL,NELEM,NETA,NLELEM,NLETA,
387.      C      4NGELSM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
388.      C      5LA(1302),LE(2002),PUN(R),
389.      C      6IPUNC,NDEGI,NQUAL,NPIW,IFEAS,IFCRSH
390.      C      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
391.      C      COMMON IA(4000),IE(8000)
392.      C      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
393.      C      COMMON/LP1/PI(1302),XX(1302)
394.      C      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
395.      C      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BP(400)
396.      C      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
397.      C      COMMON/INDX2/ JH,DIGMA,KINEAS,ICBAS
398.      C      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
399.      C      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
400.      K1= 0
401.      K2= 0
402.      IF (KMU.EQ.0) GO TO 100
403.      DO 80 K= 1,KMU
404.      K1= MU(K)
405.      IF (K1.GT.IH) GO TO 80
406.      IDD= KINBAS(K1)
407.      IF (KNU.EQ.0) GO TO 45
408.      DO 40 I= 1,KNU
409.      40 F1(K,I)= G1(IDD,I)
410.      45 F1(K)= BA(IDD)
411.      CALL UPAKC(K1)
412.      DO 60 I= 1,KMU
413.      DSUM = 0.0
414.      IF (MU(I).LE.M) DSUM= Y(MU(I))
415.      DO 50 J= 1,M
416.      IDJ= IDBAS(J)
417.      IF (IDJ.EQ.0) GO TO 50

```

```

418.      DSUM= DSUM+ Y(J)*G2(IDJ,I)
419. 50 CONTINUE
420.    D1(K,I)= DSUM
421. 60 CONTINUE
422.    BF1(K)= 0.
423.    DO 70 J= 1,M
424.      IF(IDBAS(J).EQ.0) GO TO 70
425.      BF1(K)= BF1(K)+ Y(J)*BB(IDBAS(J))
426. 70 CONTINUE
427. 80 CONTINUE
428.
429. C      NOW THE SECOND TYPE OF EQUATION IS CALCULATED BECAUSE
430. C      PI(K) IS BASIC.
431. C
432. 100 IF (KMU.EQ.0) RETURN
433.    DO 145 K= 1,KNU
434.      K2= NU(K)
435.      IF(K2.GT.1H) GO TO 145
436.      IDD= IDEAS(K2)
437.      IF (KMU.EQ.0) GO TO 123
438.      DO 120 I= 1,KMU
439.        F2(K,I)= G2(IDD,I)
440. 123 E2(K)= BS(IDD)
441.      CALL UPAKC(K2)
442.      IF (KMU.EQ. 0) GO TO 132
443.      DO 130 I= 1,KMU
444.        D2(K,I)= 0.0
445.        IF (MU(I) .LE. M) D2(K,I)= Y(MU(I))
446.      DO 125 J= 1,M
447.        IF( IDBAS(J).EQ.0) GO TO 125
448.        D2(K,I)= D2(K,I)+ Y(J)*G2(IDBAS(J),I)
449. 125 CONTINUE
450. 130 CONTINUE
451. 132 BF2(K)= 0.
452.      DO 140 J= 1,M
453.        IF (IDBAS(J).EQ.0) GO TO 140
454.        BF2(K)= BF2(K)+ Y(J)*BE(IDEAS(J))
455. 140 CONTINUE
456.      IF (K2.NE.1H) GO TO 145
457. 145 CONTINUE
458.      RETURN
459.      END
460.      SUBROUTINE FINDP(PD1,IS,P)
461. C
462. C      THIS SUBROUTINE CHOOSES THE VARIABLE TO BECOME IMPLICITLY
463. C      BASIC AS THE ONE WITH THE PIVOT ELEMENT LARGEST IN
464. C      ABSOLUTE VALUE
465. C
466.      IMPLICIT REAL*8 (A-H,O-Z)
467.      INTEGER PD,PD1,PD2,3,R,SS,RR,ZFLAG,RS,P
468.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
469.      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
470.      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
471.      COMMON/INDX2/ JH,DIGMA,KINEAS,IDBAS
472.      COMMON/DIM/ IH,N,M,KMU,KNU,MPI,NM,ITAIL
473.      COMMON/TOLFR/ TOLFZ,TOLBD,TOLCV,THETA,STPMX,STPRD
474.      P= 0
475.      IF (PD1.EQ.-1) GO TO 30
476.      IDD= KINBAS(IS)
477.      DO 20 I= 1,KNU

```



```

478.      IF (P.NE.0) GO TO 10
479.      BIG= DABS(G1(IDD,I))
480.      P= NU(I)
481.      GO TO 20
482. 10 COMP= DABS(G1(IDD,I))
483.      IF (COMP.LE.BIG) GO TO 20
484.      BIG= COMP
485.      P= NU(I)
486. 20 CONTINUE
487.      IF (BIG.LT.TOLFZ) P= 0
488.      RETURN
489. 30 IDD= IDBAS(IS)
490.      DO 40 J= 1,KMU
491.      IF (P.NE.0) GO TO 35
492.      BIG= DABS(G2(IDD,J))
493.      P= MU(J)
494. 35 COMP= DABS(G2(IDD,J))
495.      IF (COMP.LE.BIG) GO TO 40
496.      BIG= COMP
497.      P= MU(J)
498. 40 CONTINUE
499.      IF (BIG.LT.TOLFZ) P= 0
500.      RETURN
501.      END
502.      SUBROUTINE PIVOT (SS,RR)
503.  C
504.  C      THIS ROUTINE PERFORMS A PIVOT ON THE PRIMAL SUPER-
505.  C      BASIC COLUMNS IF PD.EQ.1, DUAL IF PD.EQ.-1. THE PIVOT
506.  C      BRINGS COLUMN SS INTO THE BASIS AND COLUMN RR OUT OF THE
507.  C      BASIS.
508.  C
509.      IMPLICIT REAL*8 (A-H,O-Z)
510.      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
511.      INTEGER*2 JH(350),DIGMA(952),KINRAS(1302),IDBAS(1302)
512.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
513.      DOUBLE PRECISION E(8000)
514.      REAL A(4000),C(800),CMIN,COND,ERMAX,SUMINF
515.  C
516.      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
517.      1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
518.      2NTEMP(20),KINP,ITIM,ITIM,ITINV,ITINV,MSTAT,IJOB,IROWP,IVIN,IVOUT,
519.      3ITCNT,INVERQ,ITRLIM,IFFEZ,JCOLP,NRCW,NCOL,NELEM,NETA,NLELEM,NLETA,
520.      4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
521.      5LA(1302),LE(2002),FUN(8),
522.      6IPUNC,NDAG1,NDUAL,NIPW,IFBAS,IFCRSH
523.      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
524.      COMMON IA(4000),IE(8000)
525.      COMMON/LP1/PI(1302),XX(1302)
526.      COMMON/LNCUNS/G1(350,10),G2(400,10),BA(350),BB(400)
527.      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
528.      COMMON/INDX2/ JH,DIGMA,KINEAS,ICBAS
529.      COMMON/DIM/ IH,N,M,KMU,KNU,MPI,NM,ITAIL
530.      EPS= 1E-13
531.      RR= KINRAS(RR)
532.      IF (SS.GT. IH) GO TO 5
533.      IF (NUH(SS).NE.0) GO TO 20
534. 5 CALL UNPACK(SS)
535.      CALL FTRAN(1)
536.      YTEMP(RE)= -1./Y(RR)
537.      DO 10 I= 1,M
538.

```



```

539.      IF (I.NE.RB) YTEMP(I) = Y(I)*YTEMP(RB)
540.      10 CONTINUE
541.      GO TO 30
542.      20 JS= NUH(SS)
543.      YTEMP(RB)= 1./G1(RB,JS)
544.      DO 25 I= 1,M
545.      IF (I.NE.RB) YTEMP(I)= -G1(I,JS)*YTEMP(RB)
546.      25 CONTINUE
547.      30 IF (KNU.EQ.0) GO TO 55
548.      DO 50 J= 1,KNU
549.      IF (NU(J).EQ.SS) GO TO 50
550.      V= G1(RB,J)
551.      G1(RB,J)= 0.
552.      DO 40 I= 1,M
553.      40 G1(I,J)= G1(I,J)+ V*YTEMP(I)
554.      G1(RB,J)= -G1(RB,J)
555.      50 CONTINUE
556.      55 V= BA(RB)
557.      BA(RB)= 0.
558.      DO 60 I= 1,M
559.      60 BA(I)= BA(I)+ V*YTEMP(I)
560.      BA(RB)= -BA(RB)
561.
562.      C      WE ARE NOW PIVOTING ON THE DUAL SYSTEM. IF THIS
563.      C      IS NOT AN IMPLICIT BASIC-TYPE PIVOT, WE CALL SUPERB
564.      C      TO CALCULATE THE PIVOT COLUMN.
565.      C
566.      IFL= 0
567.      RB= IDBAS(SS)
568.      IF (RR.GT. IH) GO TO 70
569.      IF (MUH(RR).NE.0) GO TO 80
570.      70 IFL= 1
571.      CALL SUPERB(1,-1,RR,0.0)
572.      JS= KNU
573.      GO TO 80
574.      80 JS= MUH(RR)
575.      82 YTEMP(RB)= 1./G2(RB,JS)
576.      DO 95 I= 1,N
577.      IF (I.NE.RB) YTEMP(I)= -G2(I,JS)*YTEMP(RB)
578.      85 CONTINUE
579.      IF (KMU.EQ.0) GO TO 115
580.      DO 110 J= 1,KMU
581.      IF (MU(J).EQ.RR) GO TO 110
582.      V= G2(RB,J)
583.      G2(RB,J)= 0.
584.      DO 100 I= 1,N
585.      100 G2(I,J)= G2(I,J)+ V*YTEMP(I)
586.      G2(RB,J)= -G2(RB,J)
587.      110 CONTINUE
588.      115 V= BB(RB)
589.      BB(RB)= 0.0
590.      DO 120 I= 1,N
591.      120 BB(I)= BB(I)+ V*YTEMP(I)
592.      BB(RB)= -BB(RB)
593.      IF (IFL.EQ.0) RETURN
594.      CALL SUPERB(2,0.0,-1,RR)
595.      RETURN
596.      END
597.
598.      C

```

```

599.      SUBROUTINE UPAKC(II)
600.      IMPLICIT REAL*8 (A-H,O-Z)
601.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
602.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(600)
603.      DOUBLE PRECISION E(8000)
604.      REAL A(4000),C(400),CMIN,COND,ERMAX,SUMINF
605.
606.      C
607.      COMMON DSUM,DPROD,DY,DE,DF,B(350),X(350),Y(350),YTEMP(350),
608.      1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
609.      2NTEMP(20),KINP,ITIM,ITIM,ITINV,ITINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
610.      3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
611.      ANGLELEM,NINF,NUELEM,NURTA,NNEGDIJ,NLINES,ISTYPE(350),
612.      SLA(1302),LF(2002),PUN(8),
613.      6IPUNC,NDEGI,NDUAL,NPIW,IFBAS,IFCRSH
614.      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
615.      COMMON IA(4000),IF(4000)
616.      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
617.      COMMON/DIM/ IH,N,M,KMU,KNU,MPI,NM,ITAIL
618.
619.      C
620.      DO 100 I= 1,M
621.      Y(I)= 0.
622.      100 CONTINUE
623.
624.      C
625.      LL= IC(II)
626.      KK= LC(II+1) - 1
627.      DO 200 I= LL,KK
628.      IR= IC(I)
629.      Y(IR)= C(I)
630.      200 CONTINUE
631.
632.      C
633.      RETURN
634.      END
635.
636.      C
637.      THE FOLLOWING ROUTINE MAKES THE CHANGES IN THE INDEX
638.      C SETS NECESSARY EVERY TIME A BASIS CHANGE IS MADE.
639.
640.      C
641.      SUBROUTINE BSCNG(S,R)
642.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
643.      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
644.      INTEGER S,SSAV,R,RSAB
645.      RSAB= KINBAS(R)
646.      JH(RSAB)= S
647.      KINBAS(S)= RSAB
648.      SSAB= IDBAS(S)
649.      DIGMA(SSAB)= R
650.      IDBAS(S)= 0
651.      IDBAS(R)= SSAB
652.      RETURN
653.      END
654.
655.      C
656.      THIS SUBROUTINE CAN DO THREE THINGS DETERMINED BY THE
657.      C PARAMETER 'KEY'. IT CAN ADD A COLUMN, REMOVE A COLUMN, OR
658.      C BOTH ADD AND REMOVE COLUMNS FROM THE PRIMAL OR DUAL SUPER-
659.      C BASIC COLUMNS, DEPENDING ON WHETHER KEY IS 1, 2, OR 0, RESP-
660.      C ECTIVELY. PD1 IS 1 OR -1 DEPENDING ON WHETHER A PRIMAL OR
661.      C DUAL COLUMN IS BEING ADDED. PD2 IS A SIMILAR FLAG FOR THE
662.      C COLUMN BEING REMOVED. IS AND JS INDICATE THE PARTICULAR
663.      C COLUMN TO BE ADDED OR REMOVED FROM THE GROUP OF COLUMNS
664.      C SPECIFIED BY PD1 AND PD2.

```

```

659.      SUBROUTINE SUPERR(KEY,PD1,IS,PD2,JS)
660.      IMPLICIT REAL*8 (A-H,O-Z)
661.      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
662.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
663.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
664.      DOUBLE PRECISION E(8000)
665.      REAL A(4000),C(900),CMIN,COND,ERMAX,SUMINF
666.
667.      C
668.      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
669.      1A,F,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
670.      2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,I0BJ,I0WP,IVIN,IVOUT,
671.      3ITCNT,INVERQ,ITRLIM,IFFEZ,JCOLP,NRCW,NCOL,NELEM,NETA,NLELEM,NLETA,
672.      4NGELEM,NINF,NUELEM,NUETA,NNEGJ,NLINES,ISTYPE(350),
673.      5LA(1302),LE(2002),PUN(8),
674.      6IPUNC,NDEGI,NDUAL,NPIW,IFEAS,IFCRSH
675.      COMMON ITC,ITCA,IFPINT,IFNEG,KOUTB
676.      COMMON IA(4000),IE(8000)
677.      COMMON/LP1/PI(1302),XX(1302)
678.      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
679.      COMMON/INDX1/ NUH(10),MLH(10),NU(10),MU(10)
680.      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
681.      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
682.      COMMON/DIM/ IH,N,M,KMU,KNU,MPI,NM,ITAIL
683.      EPS= 16.**(-13)
684.      IF (KEY.EQ.1) GO TO 50
685.      IF (PD2.GT.0) GO TO 20
686.      KM= KMU
687.      KMU= KMU- 1
688.      IF (JS .GT. 1H) GO TO 4
689.      MH= MUH(JS)
690.      MUH(JS)= 0
691.      IF (MH.EQ.KM) GO TO 6
692.      DO 5 I= MH,KMU
693.      IM= MU(I+1)
694.      MU(I)= IM
695.      MUH(IM)= MUH(IM)- 1
696.      DO 5 J= 1,N
697.      G2(J,I)= G2(J,I+1)
698.      5 CONTINUE
699.      6 MU(KM)= 0
700.      4 DO 7 J= 1,N
701.      7 G2(J,KM)= 0.0
702.      IF (KEY.EQ.2) RETURN
703.      GO TO 50
704.      20 KN= KNU
705.      KNU= KNU- 1
706.      IF (JS .GT. 1H) GO TO 28
707.      NH= NUH(JS)
708.      NUH(JS)= 0
709.      IF (NH.EQ.KN) GO TO 25
710.      DO 25 I= NH,KNU
711.      JN= NU(I+1)
712.      NU(I)= JN
713.      NUH(JN)= NUH(JN)- 1
714.      DO 25 J= 1,M
715.      G1(J,I)= G1(J,I+1)
716.      25 CONTINUE
717.      26 NU(KN)= 0
718.      28 DO 30 I= 1,M
719.      30 G1(I,KN)= 0.0

```



```

720.      IF (KEY.EQ.2) RETURN
721.      C
722.      C      NOW WE ADD THE SUPERBASIC VARIABLE SPECIFIED BY PD1,IS.
723.      C
724.      50 IF (PD1.LT.0) GO TO 70
725.      KNE= KNU
726.      KNU= KNU+ 1
727.      IF (IS.GT.IH) GO TO 55
728.      NU(KNU)= IS
729.      NUH(IS)= KNU
730.      55 CALL UNPACK(IS)
731.      CALL FTRAN(1)
732.      DO 60 I= 1,M
733.      60 G1(I,KNU)= -Y(I)
734.      RETURN
735.      C
736.      C      NOW WE ARE BRINGING A DUAL SUPERBASIC COLUMN IN.
737.      C
738.      70 KMU= KMU+ 1
739.      IF (IS.GT.IH) GO TO 73
740.      MU(KMU)= IS
741.      MUH(IS)= KMU
742.      73 IF (IS.LE.M) GO TO 110
743.      C
744.      C      THE ENTERING VARIABLE IS A ZETA. THE ENTERING COLUMN
745.      C      IS A ROW OF INV(B) AND A ROW OF INV(B)*D.
746.      C
747.      IDD= KINBAS(IS)
748.      DO 75 I= 1,M
749.      75 Y(I)= C.
750.      Y(IDD)= 1.
751.      CALL FTRAN
752.      CALL SHIFTR(3,2)
753.      DO 80 I= 1,M
754.      IF (IDBAS(I).EQ.0) GO TO 80
755.      G2(IDBAS(I),KMU)= X(I)
756.      80 CONTINUE
757.      DO 100 J= MP1,NM
758.      IF (IDBAS(J).EQ.0) GO TO 100
759.      CALL UNPACK(J)
760.      DDT= 0.0
761.      DO 90 I= 1,M
762.      IF (IDBAS(I).EQ.0) GO TO 90
763.      DDT= DDT+ Y(I)*X(I)
764.      90 CONTINUE
765.      G2(IDBAS(J),KMU)= DDT
766.      100 CONTINUE
767.      RETURN
768.      C
769.      C      THE ENTERING VARIABLE IS A PI. THE ENTERING COLUMN
770.      C      IS PARTLY C1*INV(B) AND PARTLY C2+ C1*INV(B)*D.
771.      C      -C1*INV(B) IS PART OF THE BASIS INVERSE CORRESPONDING TO
772.      C      THE X-BASIC COLUMNS AND THE T-BASIC ROWS.
773.      C
774.      110 IDD= KINBAS(IS)
775.      DO 130 I= 1,M
776.      130 Y(I)= C.
777.      Y(IDD)= 1.
778.      CALL FTRAN
779.      CALL SHIFTR(3,2)

```



```

780.      DO 140 I= 1,M
781.      IF (IDBAS(I).EQ.0) GO TO 140
782.      G2(IDBAS(I),KMU)= X(I)
783.      140 CONTINUE
784.      DO 160 J= MP1,NM
785.      IF (IDBAS(J).EQ.0) GO TO 160
786.      CALL UNPACK(J)
787.      DDT= 0.0
788.      DO 150 I= 1,M
789.      IF (IDBAS(I).EQ.0) GO TO 150
790.      DDT= DDT+ Y(I)*X(I)
791.      150 CONTINUE
792.      G2(IDBAS(J),KMU)= Y(15)+ DDT
793.      160 CONTINUE
794.      RETURN
795.      END
796.      SUBROUTINE RECALC
797.
798.      C
799.      C      THIS SUBROUTINE RECALCULATES THE SUPERBASIC COLUMNS
800.      C      USING THE CURRENT BASIS IN QR FORM. IT CAN ALSO ADD
801.      C      A VARIABLE AND COLUMN TO THE SUPERBASICS.
802.      C
803.      IMPLICIT REAL*8 (A-H,O-Z)
804.      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
805.      INTEGER*2 JH(350),DIGMA(952),KINEAS(1302),IDBAS(1302)
806.      INTEGER*2 I1TYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
807.      DOUBLE PRECISION E(8000)
808.      REAL A(4000),C(800),CMIN,CCND,ERMAX,SUMINF
809.      C
810.      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
811.      1A,E,CMIN,CCND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
812.      2NTEMP(20),KINP,ITIM,ITINV,ITINV,MSTAT,IOBJ,IRWP,IVIN,IVOUT,
813.      3ITCNT,INVERQ,ITRLIM,IFFEZ,JCOLP,NRCW,NCUL,NELEM,NETA,NLELEM,NLETA,
814.      4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,I1TYPE(350),
815.      5LA(1302),LE(2002),PUN(8),
816.      6IPUNC,NDEGI,NDUAL,NPIW,IFBAS,IFCRSH
817.      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTH
818.      COMMON IA(4000),IE(8000)
819.      COMMON/LP1/PI(1302),XX(1302)
820.      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
821.      COMMON/INDX1/ NUH(10),MUH(10),NL(10),MU(10)
822.      COMMON/INDX2/ JH,DIGMA,KINEAS,IDBAS
823.      COMMON/SCAL/ RT,NR,JJ,MFLAG,IHPI,P,PD,MPD,KFUN,KJAC
824.      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
825.      EPS= 16.**(-13)
826.      IF (KNU.EQ.0) GO TO 26
827.      DO 28 J= 1,KNU
828.      IDD= NU(J)
829.      CALL UNPACK(IDD)
830.      CALL FTRAN(1)
831.      DO 25 I= 1,M
832.      G1(I,J)= - Y(I)
833.      25 CONTINUE
834.      26 CALL SHIFTR(1,3)
835.      CALL FTRAN(1)
836.      DO 22 I= 1,M
837.      BA(I)= Y(I)
838.      22 IF (KMU.EQ.0) GO TO 140
839.      DO 130 J= 1,KMU
840.      IDD= MU(J)

```

```

841.      IF (IDD.LE.M) GO TO 70
842.      C      ZETA- VARIABLE IS ENTERING
843.      C
844.      IB= KINBAS(100)
845.      DO 35 I= 1,M
846.      35 Y(I)= 0.
847.      Y(IP)= 1.
848.      CALL BTRAN
849.      CALL SHIFTR(3,2)
850.      DO 40 I= 1,M
851.      IF (IDBAS(I).EQ.0) GO TO 40
852.      G2(IDBAS(I),J)= X(I)
853.      40 CONTINUE
854.      DO 60 K= MP1,NM
855.      IF (IDBAS(K).EQ.0) GO TO 60
856.      CALL UNPACK(K)
857.      DOT= 0.0
858.      DO 50 I= 1,M
859.      IF (IDBAS(I).EQ.0) GO TO 50
860.      DOT= DOT+ Y(I) * X(I)
861.      50 CONTINUE
862.      G2(IDBAS(K),J)= DOT
863.      60 CONTINUE
864.      GO TO 130
865.      C
866.      C      PI- VARIABLE IS ENTERING
867.      C
868.      70 IB= KINBAS(100)
869.      DO 90 I= 1,M
870.      90 Y(I)= 0.
871.      Y(IP)= 1.
872.      CALL BTRAN
873.      CALL SHIFTR(3,2)
874.      DO 100 I= 1,M
875.      IF (IDBAS(I).EQ.0) GO TO 100
876.      G2(IDBAS(I),J)= X(I)
877.      100 CONTINUE
878.      DO 120 K= MP1,NM
879.      IF (IDBAS(K).EQ.0) GO TO 120
880.      CALL UNPACK(K)
881.      DOT= 0.0
882.      DO 110 I= 1,M
883.      IF (IDBAS(I).EQ.0) GO TO 110
884.      DOT= DOT+ Y(I) *X(I)
885.      110 CONTINUE
886.      G2(IDBAS(K),J)= Y(100)+ DOT
887.      120 CONTINUE
888.      130 CONTINUE
889.      DO 150 I= 1,M
890.      Y(I)= 0.0
891.      IF (KINBAS(NM).NE.0) Y(KINBAS(NM))= 1.
892.      CALL BTRAN
893.      CALL SHIFTR(3,2)
894.      DO 160 I= 1,M
895.      IF (IDBAS(I).NE.0) BR(ICEAS(I))= X(I)
896.      160 DO 170 J= MP1,NM
897.      IF (IDBAS(J).EQ.0) GO TO 170
898.      CALL UNPACK(J)
899.      DOT= 0.0
900.

```

```

901.      DO 165 I= 1,M
902.      IF (IDBAS(I).EQ.0) GO TO 165
903.      DOT= DOT+ X(I)*Y(I)
904.      165 CONTINUE
905.      BB(IDBAS(J))= DOT
906.      170 CONTINUE
907.      RETURN
908.      END
909.      SUBROUTINE ENDPNT(JS,PD1,IS,ZFLAG,NET)
910.      IMPLICIT REAL*8 (A-H,O-Z)
911.      REAL*8 MIN,MIN2
912.      REAL*4 C(800)
913.      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
914.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
915.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
916.      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAN(10)
917.      COMMON/LP1/PI(1302),XX(1302)
918.      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
919.      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
920.      COMMON/LNCONS/G1(350,10),G2(400,10),GA(350),BB(400)
921.      COMMON/INDX1/ NUH(10),NUH(10),NU(10),MU(10)
922.      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
923.      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
924.      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
925.      COMMON/INT/ IPS(30),KDET,KOUNT,ISING,KEND
926.      COMMON/TOLR/ TOLFZ,TOLEL,TJLCV,THETA,STPMX,STPRD
927.      DIMENSION U(3),V1(10),V2(10),F(10),DOT(4),RHS(10),UL(10,10)
928.      DIMENSION G(10)
929.
930.      C
931.      C      THIS SUBROUTINE WILL FIND THE OPPOSITE ENDPPOINT OF
932.      C      G*(-1)(0) WITH RESPECT TO THE CELL DEFINED BY THE CURR-
933.      C      ENT BASIS, WHERE
934.      C      G(X,THETA)= THETA*F(X) - (1-THETA)*X
935.      C      F(X)= THE BUDGET SURPLUSES DETERMINED BY THE
936.      C      FTN AND BLCINS SUBROUTINES.
937.      C      THETA= THE HOMOTCFY PARAMETER.
938.
939.      NET= 0
940.      INFL= 0
941.      FRAC=0.5
942.      DELTA = .01
943.      TWDEL= 2.0*DELTA
944.
945.      C
946.      C      MFLAG=0 WHEN WE ARE SOLVING FOR THE QUADRATIC APPROXIMATION
947.      C      MFLAG=1 WHEN WE ARE SOLVING ONE OF THE HYPERPLANE SUBPROBLEMS.
948.      C      MFLAG=2 WHEN ONE OF THE LINEAR CONSTRAINTS IS BINDING.
949.      C      MFLAG=3 WHEN THE EQUILIBRIUM POINT APPEARS TO BE IN THE CURRENT
950.      C      CELL. NEWTON'S METHOD WILL BE IMPLEMENTED AS A TAIL ROUTINE ON F.
951.
952.      MFLAG=0
953.      KK= 0
954.      ITRY= 0
955.      IHP1= IH+ 1
956.      F(IHP1)= 0.0
957.      KMU1= KMU+ 1
958.      ICX= IHP1
959.
960.      C      INITIALIZE THE INDEPENDENT VARIABLES IN X.
961.      IF (KMU.EQ.0) GO TO 615
962.      DO 610 I= 1,KMU
963.      610 X(I)= PI(MU(I))

```



```

960.      615 IF (KNU.EQ.0) GO TO 625
961.      DO 620 I= 1,KNU
962.      KM= KMU+ I
963.      620 X(KM)= XX(NU(I))
964.      X(IHP1)= THETA
965.      ITER= 0
966.      625 ISGCT= 0
967.      5 ITER= ITER+ 1
968.      CALL GTN(J,G,X,F)
969.      CALL DERIVG(0,X,G,F)
970.      7 ISGCT= ISGCT+ 1
971.      IF (ISGCT.GT.1H) GO TO 300
972.      DO 10 I= 1,1H
973.      RHS(I)= -H(I,ICX)
974.      10 CONTINUE
975.      IF (ICX.EQ.IHP1) GO TO 35
976.      DO 24 J= ICX,1H
977.      JP1= J+ 1
978.      DO 23 I= 1,1H
979.      H(I,J)= H(I,JP1)
980.      24 CONTINUE
981.      C
982.      C      SOLVE THE LINEAR SYSTEM TO FIND THE TANGENT TO
983.      C      THE CURVE DEFINED BY G**-1(0).
984.      C
985.      35 CALL DECOMP(1H,H,UL)
986.      37 IF (ISING.NE.1) GO TO 38
987.      IF (ICX.EQ. 1) GO TO 35
988.      ICX= ICX- 1
989.      GO TO 7
990.      36 ICX= 1H
991.      GO TO 7
992.      38 CALL SOLVE(1H,UL,RHS,V2)
993.      40 DO 39 I= 1,1H
994.      IM= IHP1- I
995.      IF (IM.LT.ICX) GO TO 41
996.      IP1= IM+ 1
997.      V2(IP1)= V2(IM)
998.      39 CONTINUE
999.      41 V2(ICX)= 1.0
1000.      IF (ITER.GT.1) GO TO 45
1001.      SUM= 0.0
1002.      IF (PD.FQ.-1) GO TO 110
1003.      JB= KINBAS(JS)
1004.      IF (JB.NE.0) GO TO 104
1005.      IF (V2(1D).GT.0) GO TO 135
1006.      GO TO 140
1007.      104 DO 105 I= 1,KNU
1008.      IK= KMU+ I
1009.      SUM= SUM+ V2(IK)*G1(JB,I)
1010.      105 CONTINUE
1011.      IF (SUM.LT.0.0) GO TO 140
1012.      GO TO 135
1013.      110 JB= IDBAS(JS)
1014.      IF (JB.NE.0) GO TO 112
1015.      IF (V2(1D).GT.0) GO TO 135
1016.      GO TO 140
1017.      112 DO 115 I= 1,KMU
1018.      SUM= SUM+ V2(I)*G2(JB,I)
1019.      115 CONTINUE

```

```

1020. C
1021. C      INIT IS THE CURVE INDEX; IT TELLS US WHICH DIRECTION
1022. C      THE TANGENT MUST POINT TO KEEP GOING IN THE SAME DIRECTION.
1023. C
1024.      IF (SUM.LT. 0.0) GO TO 140
1025. 135 INIT= KDET
1026.      GO TO 45
1027. 140 INIT= -KDET
1028. 45 CALL NORM(V2,S1,IHP1)
1029.      S1= (KDET*INIT)/S1
1030.      DO 47 I= 1,IHP1
1031.          ACC(2,I)= V2(I)*S1
1032.          ACC(3,I)= X(I)
1033. 47 CONTINUE
1034.      WRITE (6,850)
1035.      DO 48 J= 1,IHP1
1036.          WRITE (6,855) F(J), (ACC(I,J),I=2,3)
1037. 48 CONTINUE
1038. C
1039. C      FIND THE FIRST BOUNDARY THAT THE TANGENT, Q(ALPHA)
1040. C      HITS. WE ARE TRYING TO FIND
1041. C      THE SMALLEST ALPHA SUCH THAT
1042. C      G(I..)*Q(ALPHA) + GB(I) = 0
1043. C
1044.      IFLAG= 0
1045.      IF (KMU.EQ.0) GO TO 249
1046.      DO 250 I= 1,M
1047.          IV= JH(I)
1048.          IF (IV.EQ.M .OR. IV.LE.IH) GO TO 250
1049.          DO 240 LL= 2,3
1050.              U(LL)= 0.0
1051.              DO 230 K= 1,KMU
1052.                  KJ= K+ KMU
1053.                  U(LL)= U(LL)+ G1(I,K)*ACC(LL,KJ)
1054. 230 CONTINUE
1055. 240 CONTINUE
1056. 241 U(3)= U(3)+ 3A(I)
1057.      IF (DABS(U(2)).LT.TOLFZ) GO TO 250
1058.      ALPHA= -U(3)/U(2)
1059.      IF (ALPHA .LT. -TJLFZ) GO TO 250
1060.      IF (ALPHA .GT. TOLRC) GO TO 244
1061.      TOD= 0.0
1062.      DO 242 K= 1,KMU
1063.          KM= K+ KMU
1064.          TOD= TOD+ G1(I,K)*ACC(2,KM)
1065. 242 IF (TOD .GT. -TOLFZ) GO TO 250
1066. 244 IF (IFLAG.EQ.1) GO TO 243
1067.      IFLAG= 1
1068.      JJ= I
1069.      MIN= ALPHA
1070.      MDD= 1
1071.      GO TO 250
1072. 243 IF (ALPHA.GT.MIN) GO TO 250
1073.      JJ= I
1074.      MIN= ALPHA
1075.      MDD= 1
1076. 250 CONTINUE
1077. 249 IF (KMU.EQ.0) GO TO 640
1078.      DO 250 I= 1,N
1079.          IF (SIGMA(I) .EQ. M) GO TO 259

```



```

1140.      WRITE (6,935) IT, (X(I),I=1,IHP1)
1141.      WRITE (6,936) S1,(G(J),J=1,IHP1)
1142.      CALL DERIVG(MFLAG,X,G,F)
1143.      IF (S1.LT.TOLCV) GO TO 280
1144.      IT= IT+ 1
1145.      NET= NET+ 1
1146.      IF (IT.LE.10) GO TO 266
1147.      MFLAG= 1
1148.      MIN= MIN/2.
1149.      IF (MIN.GT. TOLRD) GO TO 255
1150.      STOP
1151. 266 IF (NET.LT. 60) GO TO 268
1152. 269 WRITE (6,267)
1153. 267 FORMAT (' ENDPOINT HAS FAILED --- TOO MANY ITERATIONS. ')
1154.      STOP
1155. 268 CALL DECOMP(IHP1,H,UL)
1156.      IF (ISING.EQ.1) GO TO 300
1157.      CALL SOLVE(IHP1,UL,G,Z)
1158.      DO 270 I= 1,IHP1
1159. 270      X(I)= X(I)- Z(I)
1160.      GO TO 266
1161. 280 IF (X(IHP1).GT. 1.0001) GO TO 295
1162.      IF (MFLAG.EQ.2) GO TO 800
1163.      GO TO 625
1164.
1165. C      NEWTON'S METHOD FOR THE PROBLEM OF FINDING THE INTERSECTION
1166. C OF THE CURVE WITH THE CONSTRAINT DEFINED BY MPD AND JJ.
1167. C
1168. 295 MFLAG= 3
1169.      DO 100 K= 1,21
1170.          KM1= K- 1
1171.          CALL FTN(F,X)
1172.          CALL NORM(F,S1,IH)
1173.          WRITE (6,938) KM1, (X(I),I=1,IH)
1174.          WRITE (6,939) S1, (F(I),I=1,IH)
1175.          IF (S1.LT.TOLFZ) GO TO 800
1176.          CALL DERIV
1177.          NET= NET+ 1
1178.          IF (IH.GT.1) GO TO 55
1179.          IF (DABS(H(1,1)).LT. TOLFZ) GO TO 58
1180.          Z(1)= F(1)/H(1,1)
1181.          GO TO 65
1182. 55      CALL DECOMP(IH,H,UL)
1183.          IF (ISING.EQ.0) GO TO 60
1184. 58      WRITE (6,940)
1185.          STOP
1186. 60      CALL SOLVE(IH,UL,F,Z)
1187. 65      ALPH= 1.
1188.          IF (ITAIL.F).EQ.0 GO TO 80
1189.          CALL DSNT(ALPH,S1)
1190.          IF (ALPH.GT. TOLCV) GO TO 70
1191.          WRITE (6,943)
1192.          GO TO 300
1193. 70      WRITE (6,955) ALPH
1194. 80      DO 95 I= 1,IH
1195.          X(I)= X(I)- ALPH*Z(I)
1196. 95      CONTINUE
1197. 100 CONTINUE
1198.      GO TO 300
1199. 300 CALL CONCHK(GMIN,KGMIN,MP2)

```

```

1200.      THETA= X(IHP1)
1201.      IF (KMU.EQ.0) GO TO 146
1202.      DO 145 I= 1,KMU
1203.          KM= MU(I)
1204.          PI(KM)= X(I)
1205.      145 IF (KMU.EQ.0) GO TO 148
1206.      DO 147 I= KMU1,IH
1207.          IK= I- KMU
1208.          KM= MU(IK)
1209.          XX(KM)= X(I)
1210.      147 IF (GMIN.LT. -TOLRD) GO TO 170
1211.      IF (MFLAG.LE.2) GO TO 153
1212.      ZFLAG= 2
1213.      RETURN
1214.      153 IF (MFLAG.EQ.1) GO TO 150
1215.      155 IF (MPD.EQ. -1) GO TO 151
1216.      IS= JH(JJ)
1217.      PD1= 1
1218.      RETURN
1219.      151 IS= DIGMA(JJ)
1220.      PD1= -1
1221.      RETURN
1222.      150 MFLAG= 0
1223.      GO TO 625
1224.
1225.      C      IF THE CURRENT VALUE OF X VIOLATES THE CONSTRAINT DEFINED
1226.      C      BY KGMIN AND MP2, FIND A GOOD STARTING POINT FOR NEWTON'S
1227.      C      METHOD BY FINDING THE POINT ON THE LINE SEGMENT
1228.      C      ACC(1,..) + ALPHA*(X - ACC(1,..)), 0<= ALPHA<= 1
1229.      C      THAT SATISFIES THE VIOLATED CONSTRAINT EXACTLY.
1230.      C
1231.      170 JJ= KGMIN
1232.      MPD= MP2
1233.      MFLAG= 2
1234.      WRITE (6,174) MPD,JJ,GMIN
1235.      174 FORMAT(/,' THE QUADRATIC PICKED THE WRONG CONSTRAINT.',/
1236.      1,' THE MOST INFEASIBLE CONSTRAINT IS TYPE ',I3,
1237.      2,' NUMBER ',I4,/, ' WITH A VALUE OF ',F14.7)
1238.      DO 171 I= 1,IHP1
1239.          V1(I)= X(I)- ACC(3,I)
1240.      DO 175 I= 1,4
1241.          DOT(I)= 0.0
1242.      175 IF (MPD.EQ.-1) GO TO 180
1243.      DO 172 I= 1,KMU
1244.          IK= I+ KMU
1245.          DOT(1)= DOT(1)+ G1(JJ,I)*ACC(3,IK)
1246.          DOT(2)= DOT(2)+ G1(JJ,I)*V1(IK)
1247.      172 CONTINUE
1248.      STP= (-BA(JJ)-DOT(1))/DOT(2)
1249.      WRITE (6,185) G1(JJ,1),BA(JJ),DOT(1),DOT(2),STR
1250.      DO 173 I= 1,IHP1
1251.          X(I)= ACC(3,I)+ STP*V1(I)
1252.      GO TO 264
1253.      180 DO 182 I= 1,KMU
1254.          DOT(1)= DOT(1)+ G2(JJ,I)*ACC(3,I)
1255.          DOT(2)= DOT(2)+ G2(JJ,I)*V1(I)
1256.      182 CONTINUE
1257.      STP= (-BB(JJ)-DOT(1))/DOT(2)
1258.      WRITE (6,185) G2(JJ,1),BB(JJ),DOT(1),DOT(2),STR
1259.      185 FORMAT(1X,' G2, BB, DOT1, DOT2, STR :',5F13.6)
1260.

```

```

1260. DO 195 I=1, IHP1
1261. 195 X(I)=ACC(3,I)+STR*VI(I)
1262. GO TO 264
1263. 300 WRITE (6,980)
1264. 450 FORMAT(1X,'THE CURRENT QUADRATIC APPROXIMATION TO THE CURVE IS:'
1265. 1,/,1X,' F(X) A2(*U)+ A3')
1266. 455 FORMAT(1X,3F17.7)
1267. 735 FORMAT(1X,'ITERATION ',I4,' X= ',9F10.5)
1268. 936 FORMAT(1X,'NORM(G(X))= ',F11.6,' G= ',8F11.6)
1269. 940 FORMAT(1X,/'*****ALGORITHM BONDED WITH SINGULAR JACOBIAN*****')
1270. 938 FORMAT(1X,' ITERATION : ',I4,' X= ',9F10.5)
1271. 939 FORMAT(1X,'NORM(F(X))= ',F11.6,' F= ',9F10.6)
1272. 945 FORMAT(1X,'NO DESCENT POSSIBLE -- ALPHA WAS ZERO.')
1273. 955 FORMAT(1X,' STEP SIZE= ALPHA= ',F9.7)
1274. 980 FORMAT(1X,'NEWTON'S METHOD FAILED TO CONVERGE.')
1275. STOP
1276. END
1277. C QUADS FINDS THE SMALLEST NONNEGATIVE ROOT OF
1278. C U(1)*ALPHA**2 + U(2)*ALPHA + U(3) = 0, IF THERE
1279. C IS ONE. IF NOT, IMAG IS SET = 1.
1280. C
1281. SUBROUTINE QUADS(U,IMAG,ALPHA,BETA)
1282. IMPLICIT REAL*8 (A-H,O-Z)
1283. INTEGER PD,P01,P02,S,R,SS,RR,ZFLAG,RS,P
1284. INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
1285. COMMON/TOLER/ TOLFX,TOLFY,TOLGZ,TOLCV,THETA,STPMX,STPRD
1286. COMMON/DIM/ IH,N,M,KMJ,KNU,MP1,NM,ITAIL
1287. DIMENSION U(3)
1288. IMAG= 0
1289. IF (DABS(U(1)).GT. TOLFX) GO TO 10
1290. IF (DABS(U(2)).LT.TOLFY) GO TO 8
1291. ALPHA = -U(3)/U(2)
1292. BETA= ALPHA
1293. RETURN
1294. 8 IMAG= 1
1295. RETURN
1296. 10 RT= U(2)**2 - 4*U(1)*U(3)
1297. IF (RT) 20, 30, 40
1298. C
1299. C THE QUADRATIC DOES NOT INTERSECT THIS CONSTRAINT.
1300. C
1301. 20 IMAG= 1
1302. RETURN
1303. C
1304. C THERE IS A DOUBLE ROOT.
1305. C
1306. 30 ALPHA = U(2)/(U(1)*2.)
1307. BETA= ALPHA
1308. RETURN
1309. C
1310. C TWO REAL ROOTS EXIST.
1311. C
1312. 40 DIS= DSQRT(RT)
1313. ALPHA= (-U(2)-DIS)/(2.*U(1))
1314. BETA= (-U(2)+DIS)/(2.*U(1))
1315. IF (ALPHA.LE. BETA) RETURN
1316. SAVE= ALPHA
1317. ALPHA = BETA
1318. BETA= SAVE
1319. RETURN

```



```

1320.      C      END
1321.
1322.      SUBROUTINE DSENT(ALPHA,P,NORM)
1323.      IMPLICIT REAL*8 (A-H,O-Z)
1324.      COMMON/NSWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
1325.      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
1326.      DIMENSION Y(10),F(10)
1327.      DO 5 I= 1,IH
1328.      Y(I)= X(I)- ALPHA*Z(I)
1329.      CALL FTN(F,Y)
1330.      CALL NORM (F,S2,IH)
1331.      IF (S2 .LT. P,NORM) RETURN
1332.      ALPHA= ALPHA/2.
1333.      IF (ALPHA .GT. .00001) GO TO 1
1334.      RETURN
1335.      END
1336.
1337.      C
1338.      SUBROUTINE GTN(IFL,Q,U,F)
1339.      IMPLICIT REAL*8 (A-H,O-Z)
1340.      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
1341.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
1342.      COMMON/NSWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
1343.      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
1344.      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
1345.      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BR(400)
1346.      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
1347.      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
1348.      DIMENSION U(10),Q(10),F(10)
1349.      Q(IHP1)= 0.0
1350.      CALL FTN(F,U)
1351.      THETA= U(IHP1)
1352.      IF (KNU .EQ. 0) GO TO 21
1353.      DO 20 I= 1,KNU
1354.      IK= I+ KMU
1355.      QS= F(IK)+ U(IK)
1356.      Q(IK)= THETA * QS - U(IK)
1357.      20 CONTINUE
1358.      21 IF (KMU .EQ. 0) GO TO 24
1359.      DO 23 I= 1,KMU
1360.      IP= MU(I)
1361.      IDB= KINBAS(IP)
1362.      SUM= 0.0
1363.      IF (KNU .EQ. 0) GO TO 7
1364.      DO 3 K= 1,KNU
1365.      IK= K+ KMU
1366.      SUM= SUM+ G1(IDB,K) * U(IK)
1367.      3 CONTINUE
1368.      BLAM(I)= SUM + BA(IDB)
1369.      DIF= BLAM(I)
1370.      QS= F(I)+ DIF
1371.      Q(I)= THETA*QS - DIF
1372.      23 CONTINUE
1373.      24 IF (IFL .EQ. 0) RETURN
1374.      KFUN= KFUN+ 1
1375.      SUM= 0.0
1376.      IF (IFL .EQ. 2) GO TO 31
1377.      DO 25 I= 1,IHP1
1378.      SUM= SUM + ACC(2,I) * U(I)
1379.      25 CONTINUE
1380.      Q(IHP1)= SUM - RT
1381.      RETURN
1382.

```

```

1383.      30 IF (MPD.EQ.-1) GO TO 50
1384.      IF (KMU.EQ.0) GO TO 45
1385.      DO 40 I= 1,KMU
1386.      IK= I+ KMU
1387.      SUM= SUM+ G1(JJ,I)*U(IK)
1388.      40 CONTINUE
1389.      45 Q(IHP1)= SUM+ RA(JJ)
1390.      RETURN
1391.      50 IF (KMU.EQ.0) GO TO 60
1392.      DO 55 I= 1,KMU
1393.      SUM= SUM+ G2(JJ,I)*U(I)
1394.      55 CONTINUE
1395.      60 Q(IHP1)= SUM+ BB(JJ)
1396.      RETURN
1397.      END
1398.
1399.      C
1400.      C SUBROUTINE DERIVG CALCULATES THE HESSIAN OF THE
1401.      C ATTRACTION FUNCTION, PLUS SOME OTHER GRADIENT
1402.      C SPECIFIED BY IFL.
1403.      C IFL=0 : NO OTHER GRADIENT
1404.      C IFL=1 : A SPHERE DETERMINES THE GRADIENT.
1405.      C IFL=2 : ONE OF THE LINEAR CONSTRAINTS IS
1406.      C THE GRADIENT.
1407.
1408.      C
1409.      C SUBROUTINE DERIVG(IFL,U,G,F)
1410.      C IMPLICIT REAL*8 (A-H,O-Z)
1411.      C INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
1412.      C INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
1413.      C COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
1414.      C COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
1415.      C COMMON/SCAL/ RT,NR,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
1416.      C COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
1417.      C COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
1418.      C COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
1419.      C DIMENSION G(10),J(10),G(10),F(10)
1420.      C THETA= U(IHP1)
1421.      C TMTH= 1. - THETA
1422.      C CALL DERIV
1423.      C DO 20 J= 1,IH
1424.      C DO 10 I= 1,IH
1425.      C H(I,J)= THETA * H(I,J)
1426.      C
1427.      C 10 CONTINUE
1428.      C 20 CONTINUE
1429.      C DO 24 I= 1,IH
1430.      C IF (I.GT. KMU) GO TO 25
1431.      C IP= MU(I)
1432.      C IDB= KINBAS(IP)
1433.      C DO 24 J= 1,KNU
1434.      C JK= J+ KMU
1435.      C H(I,JK)= H(I,JK) - TMTH*G1(IDB,J)
1436.      C
1437.      C 24 CONTINUE
1438.      C GO TO 28
1439.      C 25 H(I,I)= H(I,I)- TMTH
1440.      C 26 CONTINUE
1441.      C DO 35 I= 1,IH
1442.      C IF (I.GT. KMU) GO TO 30
1443.      C H(I,IHP1)= F(I)+ BLAM(I)
1444.      C GO TO 35
1445.      C 30 H(I,IHP1)= F(I) + U(I)
1446.      C 35 CONTINUE

```

```

1451.      39 IF (IFL.EQ.0) RETURN
1452.      IF (IFL.EQ.2) GO TO 50
1453.      DO 40 I=1,IMP1
1454.      40 H(IMP1,I)= ACC(2,I)
1455.      RETURN
1456.      50 IF (MPD.EQ.-1) GO TO 70
1457.      DO 60 J=1,IMP1
1458.      H(IMP1,J)= 0.0
1459.      IF (J.LE.KMU) GO TO 60
1460.      IF (J.EQ.IMP1) GO TO 60
1461.      JM= J- KMU
1462.      H(IMP1,J)= G1(JJ,JM)
1463.      60 CONTINUE
1464.      RETURN
1465.      70 DO 80 J=1,IMP1
1466.      H(IMP1,J)= 0.0
1467.      IF (J.GT.KMU) GO TO 80
1468.      H(IMP1,J)= G2(JJ,J)
1469.      80 CONTINUE
1470.      RETURN
1471.      END
1472.      SUBROUTINE CONCHK(GMIN,KGMIN,MP2)
1473.
1474.      C      THIS SUBROUTINE EVALUATES THE CONSTRAINT FUNCTIONS
1475.      C      AND FINDS THE SMALLEST VALUE IN GMIN. ZFLAG IS BOTH AN
1476.      C      INPUT AND OUTPUT PARAMETER. IF ZFLAG.EQ.1 INITIALLY, THEN
1477.      C      ONLY THE NONLINEAR CONSTRAINT IS EVALUATED. ZFLAG IS SET
1478.      C      EQUAL TO 2 IF ANY CONSTRAINT IS NONPOSITIVE, OTHERWISE IT
1479.      C      REMAINS ZERO.
1480.
1481.      IMPLICIT REAL*8 (A-H,O-Z)
1482.      REAL*8 C(800)
1483.      INTEGER PD,PD1,PD2,3,R,3S,RR,ZFLAG,RS,P
1484.      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
1485.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
1486.      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAN(10)
1487.      COMMON/BLCST/RF1(10),BF2(10),E1(10),E2(10),C,IC,LC
1488.      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
1489.      COMMON/LP1/PI(1302),XX(1302)
1490.      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BR(400)
1491.      COMMON/INDX2/ JH,DIGMA,KINBAS,ICBAS
1492.      COMMON/SCAL/ BT,NB,JJ,MFLAG,IMP1,P,PD,MPD,KFUN,KJAC
1493.      COMMON/DIM/ IH,N,M,KMU,KAU,NP1,NM,ITAIL
1494.      GMIN= 10.
1495.      DO 20 I=1,M
1496.      IK= JH(I)
1497.      GG= 0.0
1498.      IF (KMU.EQ.0) GO TO 15
1499.      DO 10 J=1,KMU
1500.      JK= J+ KMU
1501.      GG= GG+ G1(I,J)*X(JK)
1502.      10 CONTINUE
1503.      GG= GG+ BA(I)
1504.      XX(IK)= GG
1505.      IF (IK.EQ.M .OR. IK.LE.IH) GO TO 20
1506.      IF (MFLAG.EQ.3 .AND. IK.EQ.NM) GO TO 20
1507.      IF (GG.GE.GMIN) GO TO 20
1508.      GMIN= GG
1509.      KGMIN= I
1510.      MP2= 1

```



```

1511.      20 CONTINUE
1512.      DO 40 I= 1,N
1513.          IK= DIGMA(I)
1514.          GG= 0.0
1515.          IF (KMU.EQ.0) GO TO 35
1516.          DO 30 J= 1,KMU
1517.              GG= GG+ G2(I,J)*X(J)
1518.      30 CONTINUE
1519.      35 GG= GG+ BB(I)
1520.          PI(IK)= GG
1521.          IF (IK.EQ. 4) GO TO 40
1522.          IF (GG.GE.GMIN) GO TO 40
1523.          GMIN= GG
1524.          KGMIN= I
1525.          MP2= -1
1526.      40 CONTINUE
1527.          RETURN
1528.      END
1529.
1530.      SUBROUTINE FTN EVALUATES THE FUNCTION F(X)=
1531.      <B(MU)+ D1*X(MU) + DIAG(X(MU))*(F1*X(MU)) >
1532.      <B(MU)+ D2*X(MU) + DIAG(X(MU))*(F2*X(MU)+ F3) >
1533.
1534.      SUBROUTINE FTN(F,Y)
1535.
1536.      THIS ROUTINE EVALUATES THE FUNCTION WHICH THE ENDPOINT
1537.      SUBROUTINE IS CURRENTLY TRYING TO FIND A ROOT OF. IF MFLAG
1538.      EQUALS 1 THEN THE LAST FUNCTIONAL COMPRISING F IS ONE OF THE
1539.      C INEQUALITY CONSTRAINTS-- THE INEQUALITY DETERMINED BY THE
1540.      PARAMETERS MPD AND JJ.
1541.
1542.      IMPLICIT REAL*8 (A-H,O-Z)
1543.      REAL*4 C(800)
1544.      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,O
1545.      INTEGER*2 JH(350),DIGMA(952),KINEAS(1302),IDRAS(1302)
1546.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
1547.      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
1548.      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
1549.      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
1550.      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
1551.      COMMON/SCAL/ BT,NR,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
1552.      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
1553.      DIMENSION F(IH),Y(IH)
1554.      KMU1= KMU+ 1
1555.      KFUN= KFUN+ 1H
1556.      IF (KMU.EQ.0) GO TO 23
1557.      DO 20 I= 1,KMU
1558.          DOT= 0.0
1559.          IF (KNU.EQ.0) GO TO 15
1560.          DO 10 J= 1,KNU
1561.              JK= J+ KMU
1562.              DOT= DOT+ F1(I,J)*Y(JK)
1563.      15 F(I)= -Y(I)*(DOT+ F1(I))+ BF1(I)
1564.          IF (KMU.EQ.0) GO TO 20
1565.          DO 10 J= 1,KMU
1566.      18 F(I)= F(I)+ D1(I,J)*Y(J)
1567.      20 CONTINUE
1568.      27 IF (KNU.EQ.0) RETURN
1569.      DO 40 I= KMU1,IH
1570.          FMP= F- KMU

```

```

1571.      DOT= 0.0
1572.      IF (KMU.EQ.0) GO TO 30
1573.      DO 25 J= 1,KMU
1574.      25 DOT= DOT+ F2(IMB,J)*Y(J)
1575.      30 F(I)= -Y(I)*(DOT+E2(IMB))+ 3F2(IMB)
1576.      IF (KMU.EQ.0) GO TO 40
1577.      DO 32 J= 1,KMU
1578.      32 F(I)= F(I)+ D2(I4B,J)*Y(J)
1579.      40 CONTINUE
1580.      RETURN
1581.      END
1582.      SUBROUTINE DERIV
1583.
1584.      C
1585.      C THIS SUBROUTINE CALCULATES THE EXACT JACOBIAN OF THE
1586.      C BILINEAR FUNCTION DEFINED BY BLCNS. THERE MAY OR MAY
1587.      C NOT BE ANOTHER FUNCTIONAL APPENDED WHICH WE ARE ATTEMPT-
1588.      C ING TO MAKE BINDING WITH NEWTON'S METHOD.
1589.
1590.      IMPLICIT REAL*8 (A-H,O-Z)
1591.      REAL*4 C(800)
1592.      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
1593.      INTEGER*2 JH(350),DIGMA(552),KINBAS(1302),IDBAS(1302)
1594.      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
1595.      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
1596.      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
1597.      COMMON/FLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
1598.      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
1599.      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
1600.      COMMON/SCAL/ RT,VE,JJ,VFLAG,IHP1,P,PD,MPD,KFUN,KJAC
1601.      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
1602.      KJAC= KJAC+ 1
1603.      DO 10 I= 1,IH
1604.      DO 10 J= 1,IH
1605.      10 H(I,J)= 0.
1606.      KMUI= KMUI+ 1
1607.      KNUI= KNU+ 1
1608.      IH= IH- 1
1609.      IF (KMU.EQ.0) GO TO 55
1610.      DO 30 J= 1,KMU
1611.      DO 30 I= 1,KMU
1612.      H(I,J)= D1(I,J)
1613.      IF (I.NE.J) GO TO 30
1614.      DOT= 0.0
1615.      IF (KNU.EQ.0) GO TO 25
1616.      DO 20 K= 1,KNU
1617.      20 DOT= DOT+ F1(I,K)*X(K)
1618.      25 H(I,J)= H(I,J)- DOT- E1(I)
1619.      30 CONTINUE
1620.      IF (KNU.EQ.0) GO TO 50
1621.      DO 40 I= KMUI,IH
1622.      IMP= I- KMUI
1623.      H(I,J)= D2(IMB,J)- F2(IMB,J)*X(I)
1624.      40 CONTINUE
1625.      50 CONTINUE
1626.      IF (KMU.EQ.0) RETURN
1627.      DO 70 J= KMUI,IH
1628.      JMUI= J- KMUI
1629.      IF (KMU.EQ.0) GO TO 72
1630.      DO 60 I= 1,KMU

```

```

1631.          H(I,J)= - X(I)*F1(I,JMU)
1632. 60      CONTINUE
1633. 70      CONTINUE
1634. 72      IF (KMU.EQ.0) RETURN
1635.          DO 80 I= 1,KMU
1636.              IK= KMU+I
1637.              IF (KMU.EQ.0) GO TO 77
1638.              DO 75 K= 1,KMU
1639. 75      H(IK,IK)= H(IK,IK)- F2(I,K)*X(K)
1640. 77      H(IK,IK)= H(IK,IK)- E2(I)
1641. 80      CONTINUE
1642.          RETURN
1643.      END
1644.      SUBROUTINE NORM(Y,S1,N)
1645.      IMPLICIT REAL*8(A-H,O-Z)
1646.      DIMENSION Y(10)
1647.      S1= 0.0
1648.      DO 10 I= 1,N
1649. 10      S1= S1+ Y(I)*Y(I)
1650.      S1= CSQRT(S1)
1651.      RETURN
1652.      END
1653.      SUBROUTINE DECOMP(NN,A,UL)
1654.      IMPLICIT REAL*8(A-H,O-Z)
1655.      COMMON/INT/ IPS(30),KDET,KCUNT,ISING,KEND
1656.      DIMENSION A(10,10),UL(10,10),SCALES(10)
1657.      N= NN
1658.      KDET= 1
1659.
1660. 00      INITIALIZE IPS,UL, AND SCALES.
1661.          ISING= 0
1662.          DO 5 I= 1,N
1663.              IPS(I)= 1
1664.              ROWNRM= 0.0
1665.              DO 2 J= 1,N
1666.                  UL(I,J)= A(I,J)
1667.                  IF (ROWNRM-DAJS(UL(I,J))) 1,2,2
1668.                      ROWNRM= DAPS(UL(I,J))
1669. 1      CONTINUE
1670. 2      IF (ROWNRM) 3,4,3
1671. 3      SCALES(I)= 1./ROWNRM
1672.          GO TO 5
1673. 4      CALL SING(1)
1674.          ISING= 1
1675.          SCALES(I)= 0.
1676. 5      CONTINUE
1677.
1678. 00      GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
1679.          NM1= N- 1
1680.          DO 17 K= 1,NM1
1681.              BIG= 0.0
1682.              DO 11 I= K,N
1683.                  IP= IPS(I)
1684.                  SIZE= DAJS(UL(IP,K))*SCALES(IP)
1685.                  IF (SIZE-BIG) 11,11,10
1686. 10      BIG= SIZE
1687.                  IXP IV= I
1688. 11      CONTINUE
1689.                  IF (BIG) 13,12,13
1690. 12      CALL SING(2)

```



```

1691.         ISING= 1
1692.         GO TO 17
1693. 13      IF (IDXP IV-K) 14,15,14
1694. 14      J= IPS(K)
1695.         IPS(K)= IPS(IDXP IV)
1696.         IPS(IDXP IV)= J
1697.         KD ET= -KD ET
1698. 15      KP= IPS(K)
1699.         PIVOT= UL(KP,K)
1700.         KP1= K+1
1701.         DO 16 I= KP1,N
1702.             IP= IPS(I)
1703.             EM= -UL(IP,K)/PIVOT
1704.             UL(IP,K)= -EM
1705.             DO 15 J= KP1,N
1706.                 UL(IP,J)= UL(IP,J)+ EM*UL(KP,J)
1707. 16      CONTINUE
1708. 17      CONTINUE
1709.         KP= IPS(N)
1710.         IF (UL(KP,N)) 19,18,19
1711. 18      CALL SING(2)
1712.         ISING= 1
1713. 19      RETURN
1714.         END
1715. C
1716. C
1717.         SUBROUTINE SOLVE(NN,UL,B,X)
1718.         IMPLICIT REAL*8(A-H,O-Z)
1719.         COMMON/INT/ IPS(30),KD ET,KCUNT,ISING,KEND
1720.         DIMENSION UL(10,10),B(10),X(10)
1721.         N= NN
1722.         NP1= N+ 1
1723.         IP= IPS(1)
1724.         X(1)= B(IP)
1725.         DO 2 I= 2,N
1726.             IP= IPS(I)
1727.             IM1= I-1
1728.             SUM= 0.0
1729.             DO 1 J= 1,IM1
1730.                 SUM= SUM+ UL(IP,J)*X(J)
1731. 1          X(I)= B(IP)- SUM
1732. C
1733.         IP= IPS(N)
1734.         X(N)= X(N)/UL(IP,N)
1735.         IF (UL(IP,N).GE.0) GO TO 10
1736.             KD ET= -KD ET
1737. 10      DO 4 IBACK= 2,N
1738.             I= NP1- IBACK
1739. C          I GUES (N-1),....1
1740.             IP= IPS(I)
1741.             IP1= I+1
1742.             SUM= 0.0
1743.             DO 3 J= IP1,N
1744.                 SUM= SUM+ UL(IP,J)*X(J)
1745.             DIV= UL(IP,I)
1746.             IF (DIV.GE.0) GO TO 4
1747.             KD ET= -KD ET
1748. 4          X(I)= (X(I)- SUM)/DIV
1749.         RETURN
1750.         END
1750.

```

```

1751.
1752.
1753.
1754.
1755.
1756.
1757.
1758.
1759.
1760.
1761.
1762.
1763.
1764.
1765.
1766.
1767.
1768.
1769.
1770.
1771.
1772.
1773.
1774.
1775.
1776.
1777.
1778.
1779.
1780.
1781.
1782.
1783.
1784.
1785.
1786.
1787.
1788.
1789.
1790.
1791.
1792.
1793.
1794.
1795.
1796.
1797.
1798.
1799.
1800.
1801.
1802.
1803.
1804.
1805.
1806.
1807.
1808.
1809.
1810.
1811.
1812.

C
SUBROUTINE SING(IWHY)
11 FORMAT(1X,'MATRIX WITH ZERO ROW IN DECOMPOSE.')
12 FORMAT(1X,'SINGULAR MATRIX IN DECOMPOSE. ZERO DIVIDE IN SOLVE')
IF (IWHY-1) 1,1,2
1 WRITE (6,11)
GO TO 10
2 WRITE (6,12)
10 RETURN
END
SUBROUTINE DEBAG(MODE)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 MIN
REAL*4 TIME
INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
INTEGER*2 ISTYPE,LA,LE,IA,IF,PUN,LC(20),IC(800)
DOUBLE PRECISION E(3000)
REAL A(4000),C(100),CMIN,COND,ERMAX,SUMINF

COMMON DSUM,DPRJD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
1A,C,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
2NTEMP(20),KINP,ITIM,ITIM,ITINV,ITINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
3ITCNT,INVERQ,ITRLIM,IFFZ,JCOLP,NKCN,NCOL,NELEM,NETA,NLELEM,NLETA,
4NGLEEM,NINF,NUELEM,NUEA,NNEGDU,NLINES,ISTYPE(350),
5LA(1302),LE(2002),FUN(3),
6IPUNC,NOLGI,NDUAL,NIPW,IFBAS,IFCRSH
COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
COMMON IA(4000),IE(8000)
COMMON/LP/PI/PI(1302),XX(1302)
COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
COMMON/SCAL/ BT,NR,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
IF (MODE-2) 10,30,60
1 TIME=1.0
WRITE(6,20) TIME
20 FORMAT (1X,' THE TIME LEFT IS NOW ',F9.6,' SEC.')
RETURN
30 WRITE (6,40)
WRITE (6,41) (BA(I),I=1,M)
WRITE (6,49) (BB(I),I=1,N)
IF (KNU,FO.0) GO TO 35
WRITE (6,42) ((G1(I,J),I=1,M),J=1,KNU)
35 IF (KMU,FO.0) RETURN
WRITE (6,43) ((G2(I,J),I=1,N),J=1,KMU)
RETURN
40 FORMAT (/,1X,' LINEAR CONSTRAINTS.')
42 FORMAT (1X,' G1= ',7F13.5)
41 FORMAT (1X,' BA= ',7F13.5)
40 FORMAT (1X,' BB= ',7F13.5)
43 FORMAT (1X,' G2= ',7F13.5)
40 IF (MODE-4) 65,70,100
65 WRITE (6,40) (KINBAS(I),I=1,NCOL)
WRITE (6,85) (JH(I),I=1,NROW)
80 FORMAT (1X,' KINBAS= ',20I4)

```

```

1813.      85 FORMAT (1X,' JH= ',2014)
1814.      RETURN
1815.      70 WRITE (6,71) KNU,KMU
1816.      WRITE (6,152) (NUH(I),I=1,IH)
1817.      WRITE (6,151) (MUH(I),I=1,IH)
1818.      WRITE (6,74) (XX(I),I=1,IH)
1819.      WRITE (6,75) (PI(I),I=1,IH)
1820.      71 FORMAT (//,' SUPEREASIC INFJ: KNU=',13,' KMU= ',13)
1821.      74 FORMAT (' T =',9F12.6)
1822.      75 FORMAT (' LAMBDA=',9F12.6)
1823.      RETURN
1824.      100 IF (MODE - 6) 101,140,140
1825.      101 WRITE (6,120) KNU,KMU
1826.      IF (KMU.EQ.0) GO TO 105
1827.      WRITE (6,121) (BF1(I),I=1,KMU)
1828.      WRITE (6,122) ((O1(I,J),J=1,KMU),I=1,KMU)
1829.      WRITE (6,123) (E1(I),I=1,KMU)
1830.      IF (KNU.EQ.0) GO TO 105
1831.      WRITE (6,124) ((F1(I,J),J=1,KNU),I=1,KMU)
1832.      105 IF (KNU.EQ.0) RETURN
1833.      WRITE (6,125) (BF2(I),I=1,KNU)
1834.      WRITE (6,126) (E2(I),I=1,KNU)
1835.      IF (KMU.EQ.0) RETURN
1836.      WRITE (6,127) ((O2(I,J),J=1,KMU),I=1,KMU)
1837.      WRITE (6,128) ((F2(I,J),J=1,KMU),I=1,KMU)
1838.      RETURN
1839.      120 FORMAT (1X,'BL CONSTRAINTS, KNU,KMU:',2I3)
1840.      121 FORMAT (1X,' BF1= ',F14.6)
1841.      122 FORMAT (1X,' O1= ',F14.6)
1842.      123 FORMAT (1X,' E1= ',F14.6)
1843.      124 FORMAT (1X,' F1= ',F14.6)
1844.      125 FORMAT (1X,' BF2= ',F14.6)
1845.      126 FORMAT (1X,' E2= ',F14.6)
1846.      127 FORMAT (1X,' O2= ',F14.6)
1847.      128 FORMAT (1X,' F2= ',F14.6)
1848.      140 WRITE (6,150) PD,KMU,KNU,IHC
1849.      WRITE (6,151) (MUH(I),I=1,IH)
1850.      WRITE (6,152) (NUH(I),I=1,IH)
1851.      150 FORMAT (//,1X,' ENDPPOINT FINDER DATA:PD,KMU,KNU,IHC= ',5I3)
1852.      151 FORMAT (1X,'MUH(I)= ',2013)
1853.      152 FORMAT (1X,'NUH(I)= ',2013)
1854.      WRITE (6,154) (JH(I),I=1,M)
1855.      WRITE (6,155) (DIGMA(I),I=1,N)
1856.      154 FORMAT (1X,'JH= ',1514)
1857.      155 FORMAT (1X,'DIGMA= ',1514)
1858.      WRITE (6,153) (XX(I),I=1,NM)
1859.      WRITE (6,156) (PI(I),I=1,NM)
1860.      153 FORMAT (1X,' XX= ',9F13.6)
1861.      156 FORMAT (1X,' PI= ',9F13.6)
1862.      RETURN
1863.      END

```


References

- [1] Elken, T.R., "On The Solution of Nonlinear Equations by Path Methods," TR SOL 77-25, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, October 1977.
- [2] Elken, T.R., "The Computation of Economic Equilibria by Path Methods," TR SOL 77-26, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, October 1977.
- [3] Forsythe, G.E., The Solution of Linear Systems of Equations, New York: Prentice Hall.
- [4] Tomlin, J.A., "Users Guide for LPML," unpublished report, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, 1975.
- [5] Tomlin, J.A., "Users Guide to LCPL -- A Program for Solving Linear Complementarity Problems by Lemke's Method," TR SOL 76-16, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, August 1976.
- [6] Tomlin, J.A., "Programmer's Guide to LCPL -- A Program for Solving Linear Complementarity Problem's by Lemke's Method," TR SOL 76-25, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, October 1976.
- [7] Wilson, R., "The Bilinear Complementarity Problem and Competitive Equilibria of Linear Economic Models," TR SOL 76-2, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, January 1976.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER SOL-78-17	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) BCA and HRA: Two Programs for Computing Economic Equilibria.		5. TYPE OF REPORT & PERIOD COVERED Technical Report	
7. AUTHOR(s) Thomas Elken		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0267 EX-76-S-03-0326	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research -- SOL Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-047-143	
11. CONTROLLING OFFICE NAME AND ADDRESS Operations Research Program -- ONR Department of the Navy, 800 N. Quincy Street Arlington, VA 22217		12. REPORT DATE August 1978	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12/132p.		13. NUMBER OF PAGES 128	
		15. SECURITY CLASS. (of this report) UNCLASSIFIED	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) This document has been approved for public release and sale; its distribution is unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Economic Equilibria Homotopy Retraction Algorithm Linear Programming Path-Following Methods Bilinear Complementarity			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report gives programmers information useful for the utilization and modification of BCA and HRA, two programs for computing economic equilibria. The BCA implements the bilinear complementarity algorithm and HRA a homotopy retraction algorithm. The report describes the subroutines and variables used as well as the usual documentation: input format, sample input and output for two problems, and program listings.			

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 765

JB

